Flexicast Network Delay Tomography

by

Earl Lawrence

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2005

Doctoral Committee:

Associate Professor George Michailidis, Co-Chair
Professor Vijayan N. Nair, Co-Chair
Assistant Professor Anna Gilbert
Assistant Professor Kerby Shedden

To Jess, thanks for waiting.

ACKNOWLEDGEMENTS

I am deeply indebted to my advisors George and Vijay. In addition to providing me with knowledge, they are most responsible for keeping me motivated and making me live up to my potential as much as possible. I take sole responsibility for any remaining unused potential.

Thank you also to the rest of my committee, Kerby Shedden and Anna Gilbert, for their helpful comments and suggestions.

More thanks than I can put down on paper go to my Uncle Les and Aunt Deb. It's hard to imagine I would have turned out this well if it weren't for them. My sister Jenny and cousins Carrie, Aaron, Laura, Mike, and Mary also contributed by putting up with me for all these years. Thank you also to the rest of my family.

Earl Lawrence plays Fender and Yamaha guitars and uses Ernie Ball and GHS strings (this is as close as I'll ever get to having liner notes).

Numerous people put up with me throughout my graduate education and each deserves some mention for their patience. For being my friend during the hardest thing I've done so far, I'd like to thank Lindsay, Catherine, Kyle, Brian, Juliet, Nick, Connie, Chris, Chrissy, James, Roger, Matt and Avo. There are of course many others, but graduate school has made me more forgetful than I'd like to admit. I apologize if you've been left off the list, you have not been left out of my heart (such eloquence for a stats guy (modesty too)). In addition, I've made many friends through our shared experience of the program. For their friendship and help, I'd like to thank Dave, Xiao, Greg, Su Bang, Maria, Danny, Akarin, and all of the rest of my fellow students. Thanks also to Derek for his guidance as in those first years.

Finally, thank you to my wife Jess who has sacrificed as much or more than I have for this pile of paper. I certainly wouldn't have made it without her.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

<u>**Table**</u>

CHAPTER I

Introduction

1.1 Background and Motivation

1.1.1 Background

Computer networks form the backbone of our information society. Over the last decade, these networks have experienced an exponential growth in terms of the number of users, the amount of traffic, and the complexity of the applications. An important feature of modern computer networks is the lack of centralized control. This has enabled service providers to develop and offer a rich variety of applications and services at different quality-of-service levels. At the same time, the decentralized nature of the environment makes it very difficult to assess network performance.

Traditional queuing and traffic models do not capture well the characteristics and complexity of network behavior (Lo Presti et al. (2002)). This has led to the emergence of *network tomography* – an area that uses *active* and *passive* traffic measurement schemes to quantify the performance and quality of service of large-scale networks. (A good review of this area can be found in Castro et al. (2004)). The performance measures of interest include link delays, dropped packet rates, and available bandwidth. This thesis focuses on estimation of link delay distributions.

We briefly summarize some relevant facts about networking that are needed for the subsequent developments. (A more detailed, yet easily accessible, explanation can be found in Marchette (2001)). Suppose one wants to transfer a file from a remote network location to the local host. The file's content is broken into pieces, called packets, that also contain origin-destination information, reassembly instructions (such a sequence numbers), and error-correcting features. The collection of packets comprising the entire file is called a flow. The origin-destination information is used by the network elements (routers and switches) to deliver the packets to the intended recipient. One can think of the routers as the intersections in a road network. Packets belonging to different flows are queued at routers, awaiting their transmission to the next router according to some protocol (first-in-first-out is common, but there are others). Physically, a queue consists of a block of computer memory that temporarily stores the packets. If the queue (memory) is full when a packet arrives, it is discarded and depending on the transmission protocol the sender may or may not be alerted. Otherwise, it waits until it reaches the front of the queue and is forwarded to the next router on the way to its destination. This queuing mechanism is responsible for observed packet losses and, to a large extent, for packet delays.

1.1.2 A Motivating Application: Voice over IP

We will briefly describe the emerging application of Voice over IP (VoIP) to motivate the issues in network tomography. Data from a VoIP study will be used to illustrate our methodology later in the thesis. VoIP or Internet telephony is a technology that turns analog voice signals into digital packets and then uses the Internet to transmit them to the intended receivers. The main difference with classical telephony is that the call does not use a dedicated connection with reserved bandwidth, but instead packets carrying the voice data are multiplexed in the network with other traffic. The quality of service (QoS) requirements in terms of packet losses and delays for this application are significantly more stringent than other non-real time applications, such as e-mail. Hence, monitoring network links to ensure that they are capable of supporting VoIP telephony is an important part of the technology. The University of North Carolina (UNC) is currently in the planning phase of deploying VoIP telephony. As part of this effort, monitoring equipment and software capable of placing such phone calls have been installed throughout the campus network. Specifically, the software allows the emulation of VoIP calls between the monitoring devices. It can then synchronize their clocks and obtain very accurate packet loss and delay measurements along the network paths.

There are 15 monitoring devices deployed in a variety of buildings and on a range of different capacity links through the UNC network. The locations include dorms, libraries, and various academic buildings. The links include large capacity gigabit links, smaller 100 megabit links, and one wireless link. Monitoring VoIP transmissions between these buildings allows one to examine traffic influenced by the physical conditions of the link and the demands of various groups of users. Figure 1.1 gives the logical connectivity of the UNC network. Each of the nodes on the circle have a basic machine that can place a VoIP phone call to any of the other endpoints. The three nodes in the middle are part of the core (main routers) of the network. One of these internal nodes, the upper router linked to Sitterson Hall, also connects to the gateway that exchanges traffic with the rest of the Internet.

The measured data consist of end-to-end delays and losses. Figure 1.2 shows the delays for the packets of one phone call (flow) between two devices. The data contain information about the entire path between a pair of end-points, which spans several links. For example, a phone call placed between the dorm and the library follows a path that goes through three main routers. Many of the features found in other types of network data can be seen here: heavy-tailed marginal distributions

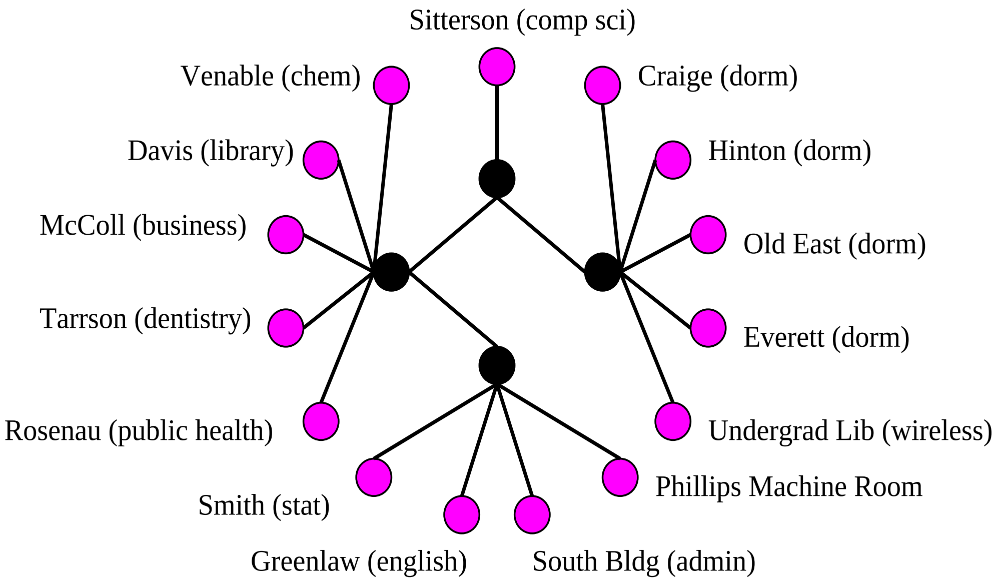

Figure 1.1: Schematic of the UNC network.

and significant autocorrelation between consecutive observations. We will see that, by using the techniques developed in this thesis, we are able to reconstruct link-level information about delays from the end-to-end path-level data.

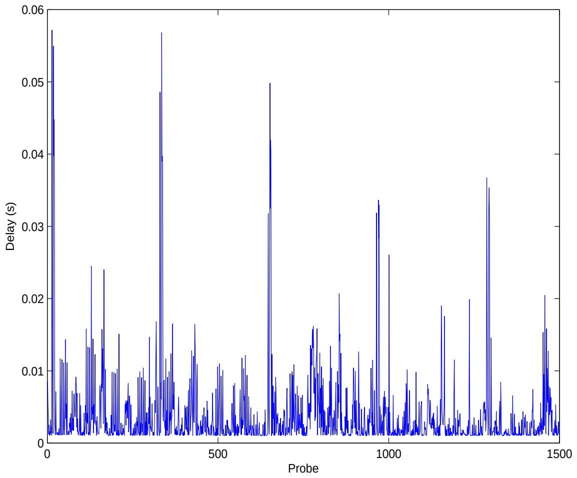

Figure 1.2: Traffic trace of packet delays generated by a single phone call across the UNC network.

1.2 Literature Review

1.2.1 Traffic Matrix Estimation

Vardi (1996) coined the phrase *network tomography* because of the similarity to the image reconstruction problem arising from positron emission tomography, a medical imaging procedure studied in the author's previous work. The goal of this and subsequent work in the traffic matrix estimation literature is to estimate the intensity of point-to-point, or origin-destination (OD), traffic between nodes in a network based upon total traffic counts observed at each link. We call this approach *passive* tomography as the measurements consist of observations made on traffic already in the network. The distinction with active tomography will be made clear in the next section.

Consider the network in Figure 1.3 from Vardi (1996). This network contains seven directed links giving seven traffic intensity measurements. There are 12 possible origin-destination pairs and thus 12 OD intensities to estimate. The problem can be stated as a linear inverse problem:

$$Y = AX, \tag{1.1}$$

where Y contains the observed link level intensities, A is the known routing matrix, and X contains the unknown OD intensities. The routing matrix has a row for each link and column for each OD pair. In the fixed routing case, $A(i, j) = 1$ if path between the OD pair j contains link i. Thus each link intensity is the sum of the OD intensities that use that link. In the random routing case, A contains probabilities. This is an inverse problem because we have measured Y and wish to make inference about X. In almost all cases, this problem is ill-posed: the number of observed link intensities n is smaller than the number of OD pairs, usually $\mathcal{O}(n^2)$. As a result, additional assumptions are necessary to regularize the problem. Vardi modeled the traffic flow as a Poisson process and sought to estimate the intensities. The Poisson assumption regularizes the problem because the variances are equal to the means so the higher order information can be used for estimation. His paper develops the maximum likelihood approach using the EM algorithm but points out certain difficulties. First, the problem becomes computationally intractable for large networks. Second, it is possible to construct simple networks for which the EM algorithm will not converge to the MLE. A moment estimator based on normal approximations is also developed resulting in a simpler algorithm.

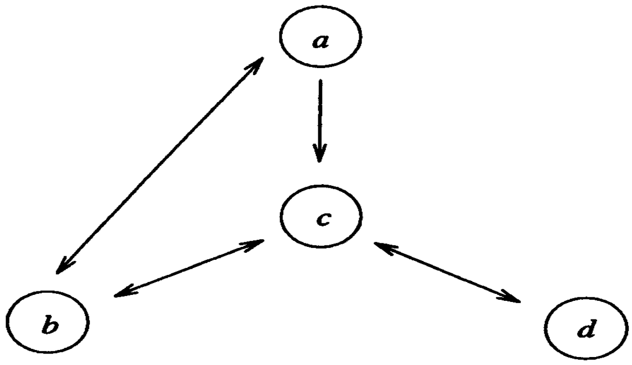

Figure 1.3: A four node network from Vardi (1996)

A Bayesian approach for the same framework was considered in Tebaldi and West (1998). The goal here is slightly different as they seek to estimate the actual OD traffic counts instead of the distribution of the counts. Again, they assume that these counts follow a Poisson process which is estimated as a means to estimate the counts. The authors present a Markov chain Monte Carlo algorithm for estimation. The article was presented with discussion and includes a somewhat critical comment by Vardi.

Cao et al. (2000) introduce a slightly different framework for the problem. They model the OD packet counts as a normal random variable in which the variance is proportional to the mean raised to a power. They specify an EM algorithm for computing the MLE. The model is extended to the situation in which the traffic intensities vary over time. This estimation is accomplished by using a moving window. Thus, all observations within a certain time period are used to estimate the parameters. The window of time is shifted one unit and the estimation is repeated independently of the previous window (despite sharing most of their observations). The results is a smoothly changing set of estimates over time. The model is fit to data collected at the Lucent gateway.

Zhang et al. (2003) present an information-theoretic approach to the traffic matrix problem. The main idea is the estimate a model that is consistent with the data while remaining as "close" as possible to an independence model. Closeness is measured in terms of entropy. In this case, the purely independent model is based on a gravity model in which traffic between s and d is a scaled product of the total traffic originating at s and the total traffic ending at d:

$$N(s,d) \approx Const N(s) N(d). \tag{1.2}$$

The model is estimated using a penalized least-squares algorithm.

1.2.2 Link Performance Estimation

The area of *active* network tomography began with Cáceres et al. (1999). Here the goal is estimate the performance characteristics of individual links based upon measurements of injected traffic sent across a network from one accessible edge node to another (or group). The challenge is to deconvolve this path-level information into the link-level information. Although it can be viewed as the reverse situation encountered in passive tomography, the active problem also gives rise to an inverse problem of the form

$$Y = AX, \tag{1.3}$$

where A is the routing matrix giving the end-to-end paths, X contains the unknown link processes, and Y gives the observed path-level data. Here the routing matrix has a row for each path and a column for each link with $A(i, j) = 1$ when link j is in the path i.

Cáceres et al. (1999) are concerned with estimating link loss rates based upon multicast probing. This paper describes a framework common in this area: tree-shaped topologies, spatially independent links, and temporally independent probes. Consider the network in Figure 1.4. A single multicast probe sent from node 0 will try to reach all of the receiver nodes: 4, 5, 6, and 7. It does so in the following manner. A packet is placed on the link from node 0 to node 1. At node 1, the packet is duplicated and sent on to the children of node 1. At each subsequent child node, the packet is duplicated further and sent. In this case, each probe generates a four-tuple observation. At each of the receivers, the packet is either received or lost. The authors assume that link losses are independent Bernoulli processes and derive an estimator based on the end-to-end measurements. The shared information resulting from the common paths and the multicast mechanism produces correlated end-to-end observations that can be used to deconvolve the path-level loss process into the link-level loss processes. This estimator involves solving a polynomial equation and the solution asymptotically corresponds to the maximum likelihood estimate.

Nowak and Coates (2001) introduce back-to-back unicast experiments for link loss estimation. Practically, this allows analysis of networks in which the multicast mechanism is not available. They develop an EM algorithm to maximize the likelihood and study some of the properties of the resulting estimator.

Active tomography for link delay inference was introduced in Lo Presti et al. (2002). The framework is similar to that used by the same authors in their link loss paper: tree topologies, multicast probing, spatio-temporal independence. Link delay is modeled with a discrete distribution, *i.e.* delay occurs in fixed units. The

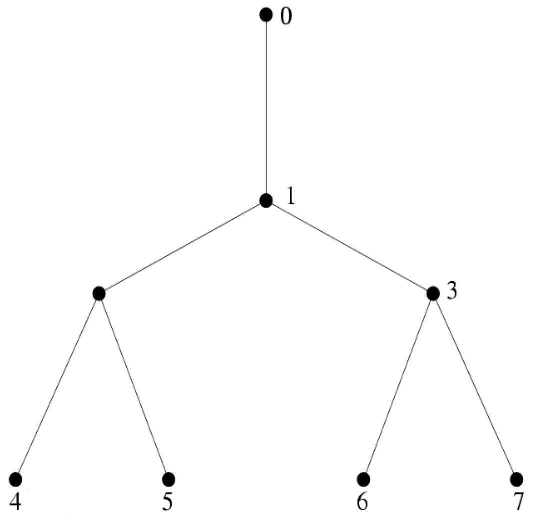

Figure 1.4: Three-layer binary tree.

estimator is similar in structure to the loss estimator by the same authors: each probability is a solution to a polynomial equation. In the delay case, this solution is not the MLE asymptotically or otherwise.

Padmananhan et al. (2003) describe a method of link loss performance estimation based on passively collected end-to-end data from client-server exchanges. They present methods of analyzing this data using random sampling, linear optimization, and Gibbs sampling.

Liang and Yu (2003) present a pseudo-likelihood method appropriate for both link performance and traffic matrix estimation. The basic focus is to consider pairwise projections of the high-dimensional observation. The result is an algorithm with improved computational efficiency at the expense of some statistical efficiency. Some important properties are preserved, namely consistency.

Tsang et al. (2003) estimate link delay distributions based on back-to-back unicast measurements. They alter the traditional discrete model slightly and introduce an efficient EM algorithm based on the fast Fourier transform.

Shih and Hero (2003) also model delay using back-to-back unicast probing. They model each link delay distribution as a finite mixture of Gaussians with a point mass at zero. They use a penalized maximum likelihood approach to choose the number of mixing components and estimate the parameters.

1.2.3 Topology Discovery

A third area of network tomography is concerned with discovering devices on a network. There are deterministic tools like *traceroute* that report the network devices and their connectivity. Unfortunately, many routers do not identify themselves to these types of tools. As a result, some authors have developed probing-based techniques for identifying the network topology. The literature, Duffield et al. (2002), Coates et al. (2002), Rabbat et al. (2004), and Shih and Hero (2004), includes clustering, maximum likelihood, and Bayesian techniques. Additionally, Coates et al. (2002) introduces the clever sandwich probe consisting of two small probes bound for receiver A with a large probe between them bound for receiver B. The time difference between the two small probes is used as a metric on the length of the shared path between the two probes. Refer to the citations for details.

1.3 Overview and Contributions

The subject of this thesis is the estimation of link delay distributions based upon end-to-end probing. The next chapter lays out the estimation framework in terms of topology, probing, and stochastic assumptions.

Chapter III develops the estimation for a discretized delay estimator. Although the framework has been considered before, this research makes several contributions. First, the question of identifiability is studied in detail including specific requirements for network experimentation that lead to estimability of the model. Further, the maximum likelihood estimator is rigorously developed. Implementation issues, asymptotic properties, and numerical performance are all considered. In addition, we consider a fast algorithm based on the idea of local maximum likelihood. Finally, the estimators are applied to real data, something of a novelty in the current literature.

Chapter IV considers link delay estimation in the more natural and general setting of continuous link delays. This approach has not be well studied due to technical difficulties. Again, we consider identifiability conditions. In this setting, we consider probing requirements as well as distributional conditions. We briefly consider maximum likelihood estimation and discuss the difficulties associated with it. We systematically develop moment estimation based on least-squares. For this general procedure, we consider several models and examine algorithmic, inferential, and numerical properties. In addition to parametric modeling, we also adopt semiparametric approach in which a few moments are specified, such as

$$E(X_k) = \mu_k \tag{1.4}$$

$$Var(X_k) = \phi\mu_k^\gamma. \tag{1.5}$$

This allows to us to make as few shape restrictions as necessary and still estimate the quantities of interest. Both approaches can be extended by including point masses at zero and infinity and the estimation procedure can be modified to account for this type of mixture.

CHAPTER II

Framework

In this chapter, we describe a framework for the estimation of link delay distributions based upon end-to-end path measurements. The three key features are the network topology, the probing mechanism, and the stochastic assumptions. The first two are discussed in detail while we give an overview of the third. Chapters III and IV develop further details of the stochastic models.

We begin by describing the inverse nature of the problem in detail. We will be estimating link delays based on active end-to-end probing. Thus, we will observe sums of link delays. The relationship between the observed path delays can be given by the following matrix equation

$$Y = AX \tag{2.1}$$

where Y contains observed path delays, X are the unobserved link delays, and A is the routing matrix with a column for each link and row for each path and $A(i, j) = 1$ is path i includes link j. Because of the nature of the topologies and the fact that the internal nodes are inaccessible, there will be more links than paths and A will be noninvertible. As an example, consider the simple topology in Figure 2.1 where node 1 is inaccessible, and nodes 2 and 3 are probed from node 0. We have

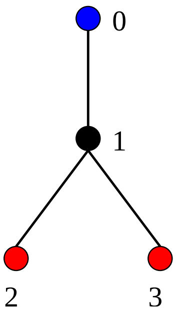

Figure 2.1: Simple topology.

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}. \tag{2.2}$$

Estimating the distribution of the X process from the observed Y process is a statistical inverse problem. In order to solve the problem, we will have to consider some sort of regularization. This is the statistical challenge of the problem. This will be solved by choosing appropriate models and using clever probing.

There are some mathematical similarities with the passive tomography problem introduced by Vardi and described earlier. In that problem, observations of packet counts are made at each link. The observed counts at each link are the sums of all the path-level traffic flows that utilize the link. The sums are also determined by a routing matrix similar to that described above. The regularization and solution, however, take quite different forms.

2.1 Topology

Throughout this thesis, networks will be described as graphs with the nodes representing computers, routers, hubs, and other physically connected devices and the edges representing communication links between devices. In this work, we will focus on *single-rooted tree topologies.* For the following, refer to Figure 2.2 as an example. Formally, let $\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$ be a tree network with node set $\mathcal{V}$ and link set $\mathcal{E}$. The node

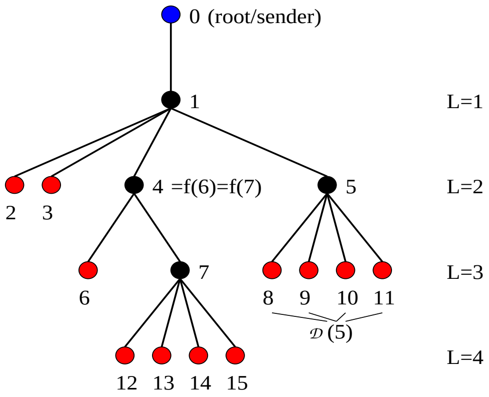

Figure 2.2: Example of a tree topology with notation.

set can be decomposed: $\mathcal{V} = \{0, \mathcal{I}, \mathcal{R}\}$. The root node 0 will be the source of all probe traffic on the network. The receiver nodes $\mathcal{R}$ are the destination for all probe traffic. The internal nodes $\mathcal{I}$ are interior points of the network that are inaccessible in the current setting. Nodes will be numbered canonically moving left to right and down the tree. Each member of $\mathcal{E}$ is a directed link named after the node at its terminus. The root node 0 will connect to a single member of $\mathcal{I}$ labeled 1 along a directed link from 0 to 1. Each member of $\mathcal{I}$ will have a single incoming link and at least two outgoing links. Each member of $\mathcal{R}$ will only connect to a single incoming link. Consider a connected pair of nodes j and k with a directed link from j to k. Node j will be called the parent of k and denoted as $f(k) = j$. For further ancestry relationships, define $f^i(k) = f(f^{i-1}(k))$ with $f^1(k) = f(k)$. A node k is said to be in layer L if $f^L(k) = 0$. The group of nodes with parent k will be denoted $\mathcal{D}(k)$, the children of k. Let $\mathcal{P}_{i,j}$ denote the path between any two connected nodes i and j.

There are many situations in which it is convenient to consider *binary* trees. A binary tree is a special case of the tree topology in which each internal node has exactly two children. A further special case is the *symmetric* binary tree (Figure 2.4 which is a binary tree in which all of the members of the receiver group $\mathcal{R}$ are in the same layer.

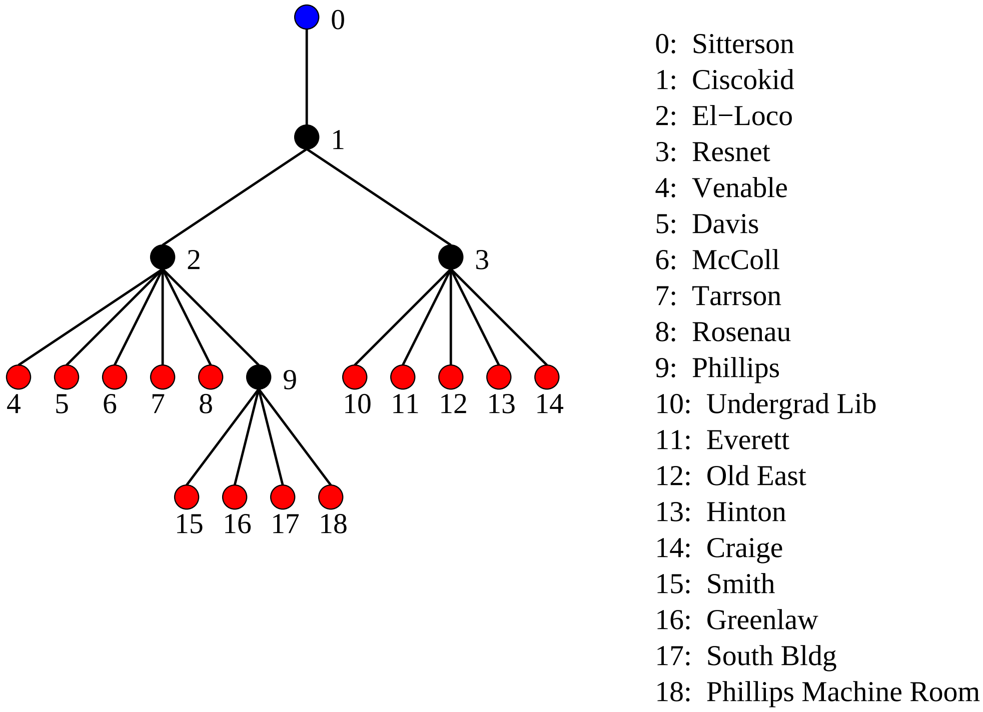

Figure 2.3: UNC Network with Sitterson Hall as the root node.

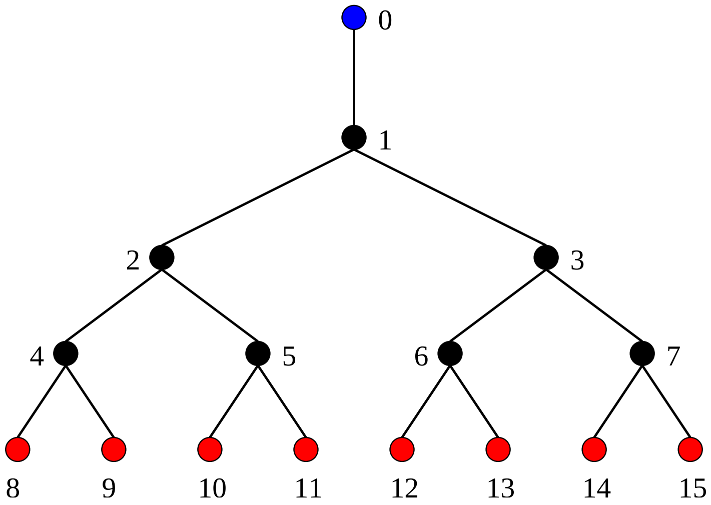

Figure 2.4: Four layer binary, symmetric tree.

There is a distinction to be made between the logical topology and the physical topology of a network. In our framework, each internal node must have at least three connections: to its parent node and to at least two children nodes. In other words, to appear in the logical topology, an internal node must be a branching point. There may be a physical device in the interior of the network at which no branching occurs;

i.e. it connects to its parent and only one child, but we will not consider this as a node on our tree (as this constitutes a "chain" and the link-level information cannot be separated). Both of the links above and below this device are considered to be part of a single logical link.

2.2 Probing Mechanism

We will consider the use of two types of protocols for probing a network. The first, called *multicast*, provides a way to probe groups of receivers simultaneously. As described in Chapter I, Section 1.2, a multicast probe begins with one packet on the first link. At each internal node, the packet is duplicated and sent to each of the node's children that are required to reach the desired receivers. Consider the tree in Figure 1.4. A packet is placed on the link from 0 to 1. At node 1, the packet is duplicated. One copy is sent to node 2 and another is sent to node 3. At node 2, the packet is duplicated and sent to 4 and 5. At node 3, the packet is duplicated and sent to 6 and 7. If we were to measure end-to-end delay of this multicast probe, we would observe a correlated four-tuple. A portion of each of the four delays would result from the delay experienced on link 1. A portion of the delays observed at nodes 4 and 5 would result from the delay experienced on node 2. Likewise for nodes 6 and 7. The correlation caused by the shared paths will be the key to deconvolving the path-level information. We note here that previous literature uses the term multicast for the protocol and the scheme by which all the receivers are probed simultaneously. We will refer to this scheme as omnicast in order to avoid confusion.

In some networks, multicast probing may be unavailable for either technical or security reasons. When this is the case, we will rely on back-to-back unicast probing. This type of probing uses the more common unicast traffic. Consider probing just the pairs $\langle 4, 5\rangle$ on Figure 1.4. We can send two unicast packets, one bound for node 4 and the other for node 5, very closely together in time. The gap between placing these packets onto link 1 should be on the order of the machine's smallest unit of time. Ideally, these probes will enter the queues on the shared links without any other traffic arriving between them. The result is that the two packets experience identical network conditions along the shared paths and any delay experienced will be identical for both probes. This allows us to mimic the correlation that we observe from the multicast protocol. For the purposes of our study, we will assume that the delays and losses along the shared paths are perfectly identical.

We will probe the network using groups of receivers of different sizes. We call this *flexicast* probing. As an example, consider the network shown in Figure 2.2. We could probe the network using any of the following collections of receiver groups: $\{\langle 2, 3\rangle, \langle 6, 12\rangle, \langle 13, 14, 15\rangle, \langle 8, 9, 10, 11\rangle\}$, $\{\langle 2, 3\rangle, \langle 6, 12\rangle, \langle 13, 14\rangle, \langle 8, 15\rangle, \langle 9, 10\rangle, \langle 11\rangle\}$, $\{\langle 2, 3, 6, 8, 9, 10, 11, 12, 13, 14, 15\rangle\}$. Breaking the receivers into groups and using flexicast probing has several advantages. First, it reduces the complexity of the data. The omnicast scheme on this tree results in 11-tuples. The more flexible scheme using pairs and singletons can cover the whole tree with five pairs and one singleton. This data is considerably less complex in terms of storage and processing. We will see later that flexible receiver groups lead to better computational complexity for the estimation procedures. Finally, these groups allow the network to be investigated more intelligently. Different parts of the network can be probed with different intensities and at different times. Sensitive parts of the network can be probed as lightly as necessary to avoid disturbance.

2.3 Stochastic Assumptions

Let $X_k(t)$ be the delay accumulated along link k from probe t. Although we will consider several distributional assumptions for the link delays, we will always assume spatial and temporal independence of the following form:

$$Cor(X_k(s), X_k(t)) = 0, \tag{2.3}$$

$$Cor(X_k(t), X_l(t)) = 0. \tag{2.4}$$

The delay on link k for probe t is completely independent of the delay on link k for any other probe s. Further, the delay on link k for probe t is completely independent of the delay on any other link l for probe t. This is a simplifying assumption. In practice, the data typically display some amount of both types of correlation. For the remainder of this discussion, we will drop the notational dependence on the probe unless it is necessary. Let Y_k be the cumulative delay accumulated on $\mathcal{P}_{0,k}$, the path from 0 to k. Observations will consist of $Y_r,\ r \in \mathcal{R}$, cumulative delays made at the receiver nodes.

We will consider two modeling paradigms that will be discussed in more detail in the appropriate chapters. We give a brief summary here.

One approach will be a discrete delay model based on discretizing the delay into units. The delay on each link will be multinomial with a certain probability of seeing i units of delay. Although artificial, this model has several nice properties. First, it makes no assumptions with regard to shape: heavy tails and massive probability on the zero unit bin are easily handled. Further, the model is quite flexible with regard to modeling goals: large units can be used to study tail behavior and small units can used to observe detailed approximations of the distributions.

A second more appropriate approach is model link delay without imposing the discrete framework. This presents several challenges. Practically, most parametric distributions are inappropriate for detailed modeling. We will discuss parametric fitting as the approach does have merit for understanding basic aspects of the underlying distributions. We will also investigate semiparametric techniques in which a few moments are specified with no further shape restrictions.

CHAPTER III

Discrete Delay Modeling

This chapter considers delay tomography for nonparametric discrete link delays. This is an approximation to the true setting, but it is a useful one. We shall see that this model is useful in a number of situations, especially for detailed link modeling.

3.1 Stochastic Model

Following Lo Presti et al. (2002) and Liang and Yu (2003), we will consider nonparametric estimation of the delay distributions using a discrete distribution with a fixed, universal bin size. Although it seems restrictive, this framework is useful for a number of reasons. First, experience with real network traffic data shows that the behavior tends to vary with the particular network being studied. So selecting a particular parametric family for modeling the link delay distribution is difficult. Moreover, the delay data typically exhibit bursty behavior, in which case the tails of the distribution are also of considerable interest. The discrete model makes no assumptions about where the mass is located, so the tails of heavy-tailed distributions can be easily estimated given a sufficient number of probes. Finally, unlike parametric families that may not be closed under convolution, the discrete delay model can easily handle cases where one logical link is actually a “chain” consisting of several physical links.

The bin size can be chosen adaptively after the data are collected. Smaller bin sizes can be used to estimate detailed information about the distribution. Large bin sizes can be used to obtain tail information. Examples of this will be discussed in the data analysis section.

Let X_k be the delay accumulated on link k, taking values in the set $\{0, q, \ldots, bq\}$. Here q is the bin size and b is the maximum discrete delay (assumed to be common for all the links). In this paper, we will ignore losses or infinite delays. One can always estimate the loss rates and the finite delay distributions separately and combine the results to estimate the overall network behavior. We make the assumption (common in the network tomography literature) that the packet delays are temporally independent and that the delay of a packet on a link is independent of the delay on the other links in the path. The assumption of temporal independence is reasonable as long as the interval between probes is large enough. Temporal stationarity is reasonable as long as the probing period is short enough to avoid major network changes. The adequacy of the spatial assumption will depend on the particular network being studied and whether there are other physical links connecting the nodes.

The data are collected by recording the total delay that a packet experiences as it travels from the root node to the receiver nodes. If the current scheme has a collection of k receivers, a single observation would be a k-tuple of delays. For example, in Figure 2.2, probe packets would be sent from node 0 to various collections of nodes 2, 3, 6, 8, 9, 10, 11, 12, 13, 14, and 15 and the delays experienced along their corresponding paths would be recorded. Physically, each end-to-end delay is the sum of the individual link delays along the path.

Let $\mathcal{P}_{0,k}$ denote the path from node 0 to node k, and let $Y_k = \sum_{i \in \mathcal{P}_{0,k}} X_i$ be the cumulative delay accumulated from the root node to node k. For example,

$Y_3 = X_1 + X_3$ in Figure 2.2. The measurements obtained from a delay tomography experiment consist of cumulative delays $Y_r,\ r \in \mathcal{R}$. Let $\alpha_k(i) = \mathrm{P}\{X_k = iq\},\ i = 1, ..., b$. Our objective is to estimate this set of values for $k \in \mathcal{E}$ and i in $\{0, 1, \ldots, b\}$ using the Y_r measurements.

In the following, we will use the notation $\vec{\alpha}_k = [\alpha_k(0), \alpha_k(1), \ldots \alpha_k(b)]'$ and $\vec{\alpha} = [\vec{\alpha}_0', \vec{\alpha}_1', \ldots, \vec{\alpha}_{|\mathcal{E}|}']'$. Let $\pi_{j,k}(i)$ be the probability that the delay accumulated on path $\mathcal{P}_{j,k}$ is equal to i units. This is a function of $\vec{\alpha}$.

3.1.1 Identifiability

It is clear from the earlier discussion that an experiment based on combinations of independent unicasts will not be able to identify all the link level parameters. So the natural question is: When will a flexicast experiment lead to identifiability? We provide below a necessary and sufficient condition. The proof is based on the idea that an individual k-cast probe can identify all of the paths between branching nodes on its subtree. It suffices for the collection to be rich enough in terms of subtrees, that the individual links can be expressed as functions of paths from different schemes. We formalize this intuition in the following Proposition.

Proposition III.1. *Let $\mathcal{T}$ be a general tree network, and suppose its link delay distributions are discrete. Let $\mathcal{C}$ be a collection of k-cast schemes $\mathcal{C}_j,\ j = 1, ..., M$. The link-level delay distributions are identifiable if and only if: (a) For every internal node $s \in \mathcal{T} \backslash \{0, \mathcal{R}\}$, there is at least one k-cast scheme $\mathcal{C}_j \in \mathcal{C}$, with $k > 1$, such that s is a branching node for $\mathcal{C}_j$, and (b) every receiver $r \in \mathcal{R}$ is covered by at least one $\mathcal{C}_j \in \mathcal{C}$.*

Remark 1: We have restricted attention to discrete distributions as they are the focus of the present paper, but the result holds more generally. First, the result can be shown to hold as long as the distribution has at least one point mass. It will also hold for purely continuous distributions under some conditions (such as higher order moments depending on the mean). It does not, however, hold for arbitrary continuous distributions. This can be seen using a two-layer tree with a source node 0, internal node 1 and receiver nodes 2 and 3. Let the link-level delay random variables be X_1, X_2, and X_3 and the path-level delay random variables at receiver nodes 2 and 3 be $Y_2 = X_1 + X_2$ and $Y_3 = X_1 + X_3$. Suppose the X_i's are independent $\mathrm{N}(\mu_i,\ 1)$. Then, it is easy to see that we cannot recover the μ_i's from the joint distribution of Y_2 and Y_3.

Remark 2: The minimum identifiable flexicast experiment in terms of data and algorithm complexity is based on a collection of bicast and unicast schemes. It is minimum in the sense that it leads to at most two-dimensional measurements. A simple way to construct such experiments is to select bicast schemes to branch at every internal point and to cover as many distinct receivers as possible and use unicast schemes to cover the remaining receivers. For the tree in Figure 2.2, one choice of a minimum identifiable experiment is the following: bicast probes to pairs $\langle 2, 3\rangle$, $\langle 6, 12\rangle$, $\langle 13, 14\rangle$, $\langle 8, 9\rangle$ and unicast probes to $\langle 10\rangle, \langle 11\rangle$, and $\langle 15\rangle$. However, in terms of reducing the total amount of probe traffic, a more efficient flexicast experiment consists of bicast probes to $\langle 2, 3\rangle$, $\langle 6, 12\rangle$, 3-cast to $\langle 13, 14, 15\rangle$, and 4-cast to $\langle 8, 9, 10, 11\rangle$. In practice, the choice of the particular flexicast experiment will be dictated by different practical considerations. In fact, one can change the combination of k-cast schemes over time and also probe different areas with varying degrees of intensity depending on the congestion or other issues that may arise.

Remark 3: An important feature of the identifiability condition is that it allows a user to plan a probing experiment that is guaranteed to estimate all of the delay distributions regardless of their form. This is a key contribution that is new in the literature. In contrast, previous conditions depend on the form of the information matrix or parameters of the distribution which are not known in advance.

Proof: The proof will proceed as follows. We first show sufficiency. We start by establishing the sufficiency of an omnicast experiment. This will be used to show that an individual k-cast scheme identifies all of the distributions on the paths between source, branching nodes, and receivers of its subtrees. With this fact, we can show that the above conditions guarantee that we have enough subtrees that the estimated collection of paths can be used to solve for every link delay distribution. The proof of necessity will then proceed by contradiction.

For omnicast probing, we consider two cases.

Case 1: Let k be some receiver node. Consider all omnicast probes that result in zero delay on all of the receivers except k. This set of probes will allow us to estimate $\vec{\alpha}_k$ as these probes consist of direct observations from link k. This argument applies to all receivers.

Case 2: We proceed by induction. Consider some node k that is not a receiver. Assume that we have identified all of the distributions for all of the links that are descended from k. Let $\mathcal{R}(k)$ represent the receivers descended from k. Consider probes that result in zero delay on all nodes except those in $\mathcal{R}(k)$ which all experience i delay. Let $\gamma^k(i)$ be the probability that each $r \in \mathcal{R}(k)$ has an end-to-end delay of i. With the described probes, we will be able to estimate $\gamma^k(i)$ for each i. From these values and the estimates for the link delay distributions of the descendants of k, we can estimate $\vec{\alpha}_k$.

This proof implies that a single k-cast scheme will identify the following distributions: the path between the source and and the first splitting node, the paths between any two splitting nodes, and the paths between a splitting node and a receiver.

Now we focus on a collection of flexicast schemes. Here we consider three cases.

Case 1: There is some k-cast scheme $\mathcal{C}_j$ in which branching occurs at node 1, the only child of the root node. Based on the omnicast identifiability proof, this experiment identifies the delay distribution for link 1, $\vec{\alpha}_1$.

Case 2: Let s be some internal node. Assume that we have identified all of the delay distributions for links $k \in \mathcal{P}_{0,f(s)}$. There is a scheme $\mathcal{C}_j$ for which branching occurs at node s. This scheme identifies the path-level distribution $\vec{\pi}_{0,s}$. We can construct $\vec{\pi}_{0,f(s)}$ and solve for $\vec{\alpha}_s$:

$$\begin{aligned}
\alpha_s(0) &= \pi_{0,s}(0)/\pi_{0,f(s)}, \\
\alpha_s(d) &= \frac{1}{\pi_{0,f(s)}(0)}\left[\pi_{0,s}(d) - \sum_{\delta=\max(0,d-B_{f(s)})}^{d-1} \alpha_s(\delta)\pi_{0,f(s)}(d-\delta)\right] \; \forall d = 1,\ldots,b,
\end{aligned}$$

where $B_{f(s)}$ is the maximum delay up to this node. We call this solution peeling since we are peeling the unknown distribution from the path-level distributions. It can be used more generally and take other functional forms.

Cast 3: Let r be some receiver node. Assume that we have identified all of the delay distributions for links $k \in \mathcal{P}_{0,f(r)}$. There is some scheme $\mathcal{C}_j$ which probes receiver r. From this, we can estimate the path probability $\pi_{0,r}$. We can construct $\vec{\pi}_{0,f(r)}$ and use peeling to get $\vec{\alpha}_r$.

It is easy to see the necessity of covering all of the receivers: if we do not probe a receiver, we can never estimate its link delay distribution.

To see the necessity of branching at each internal node, consider a collection of schemes in which branching occurs at all internal nodes except some node s. Each link $d \in \mathcal{D}(s)$ will always occur as part of a logical link that also includes link s. We will be able to obtain estimates for $\vec{\pi}_{f(s),d}$ for each $d \in \mathcal{D}(s)$ but we will have no information with which to peel the two apart. In essence, these estimates are like unicast measurements which are not sufficient for estimation. □

3.2 Maximum Likelihood Estimation

Inference for active delay tomography is an instance of a large-scale inverse problem. For example, consider the omnicast problem for the topology given in Figure 2.2. Here we must use the 11-dimensional end-to-end measurements to estimate the 15 link delay distributions. The key here is the dependence among the 11-dimensional data induced by the simultaneous probing. This dependence give us additional information that allows us to deconvolve the path-level delay into link-level information. We consider maximum likelihood estimators here. Other heuristic schemes that are computationally faster will be considered later.

First we need some additional notation. Let $\mathcal{T}^{\mathcal{C}_j}$ be the subtree probed by scheme $\mathcal{C}_j \in \mathcal{C}$, with node set $\mathcal{V}^{\mathcal{C}_j}$ and link set $\mathcal{E}^{\mathcal{C}_j}$. Let $\mathcal{X}^{\mathcal{C}_j} = \{0, 1, \ldots, b\}^{|\mathcal{E}^{\mathcal{C}_j}|}$ be the set of all possible link delay combinations that could arise from this scheme. Each $x \in \mathcal{X}^{\mathcal{C}_j}$ is an $|\mathcal{E}^{\mathcal{C}_j}|$-tuple giving a possible link-delay combination. Let the function $y(x, \mathcal{T}^{\mathcal{C}_j})$ give the end-to-end delay arising in scheme $\mathcal{C}_j$ from link outcome $x \in \mathcal{X}^{\mathcal{C}_j}$. Define the set of all possible end-to-end delays as $\mathcal{Y}^{\mathcal{C}_j} = \{y(x, \mathcal{T}^{\mathcal{C}_j}) | x \in \mathcal{X}^{\mathcal{C}_j}\}$. Let $\gamma_{\mathcal{C}_j}(y) = P\{Y^{\mathcal{C}_j} = y\}$, the probabilities for the end-to-end experimental outcomes.

Let us illustrate this notation using Figure (2.2). Suppose we probe the pair $\langle 2, 3 \rangle$. Let $b = 1$ so $X_k \in \{0, 1\}$. The link set is $\mathcal{E}^{\langle 2,3 \rangle} = \{1, 2, 3\}$. Assume that only a single probe packet is used for this scheme, and it experiences link delays of 0, 1, and 1 on each link, respectively. We then have $x = (0, 1, 1)$ and $y = (1, 1)$. The probability of this link delay set is $\mathrm{P}\{X^{\langle 2,3 \rangle} = (0, 1, 1)\} = \alpha_1(0)\alpha_2(1)\alpha_3(1)$. The probability of this end-to-end outcome is $\mathrm{P}\{Y^{\langle 2,3 \rangle} = (1, 1)\} = \alpha_1(0)\alpha_2(1)\alpha_3(1) + \alpha_1(1)\alpha_2(0)\alpha_3(0)$

which is the sum of the probabilities for the link outcomes which can give rise to this end-to-end outcome.

3.2.1 EM Algorithm

The discrete nonparametric modeling framework results in multinomial outcomes for the path-level data. Specifically, the observations consist of the number of times that one observes each individual outcome $\vec{y}$ from the set of outcomes $\mathcal{Y}^{\mathcal{C}_j}$ for a given scheme. We denote these counts $N_{\vec{y}}^{\mathcal{C}_j}$. Consider the likelihood equation:

$$l(\vec{\alpha};\mathbf{Y}) = \sum_{\mathcal{C}_j \in \mathcal{C}} \sum_{\vec{y} \in \mathcal{Y}^{\mathcal{C}_j}} N_{\vec{y}}^{\mathcal{C}_j} \log[\gamma_{\mathcal{C}_j}(\vec{y};\vec{\alpha})]. \tag{3.1}$$

This equation is difficult to maximize directly. However, it is a classical example of a missing data problem: if the counts for the unobserved link delays were known, the maximization would be fairly straightforward as the link outcomes are also simple multinomial experiments. The EM algorithm is a natural candidate for computing the maximum likelihood estimates. We just need to impute the sufficient statistics for each link: the counts for the number of times that X_k took on each value.

The E-step can be broken into two parts. Assume that we have some parameter vector $\vec{\alpha}^{(q-1)}$. First, we compute the expected number of times each link delay vector, $\vec{x}$, occurred as

$$N_{\vec{x}}^{\mathcal{C}_j\ (q)} = \frac{P\{\vec{X}^{\mathcal{C}_j} = \vec{x}\}^{(q)}}{P\{\vec{Y}^{\mathcal{C}_j} = \vec{y}(\vec{x})\}^{(q)}} N_{\vec{y}}^{\mathcal{C}_j}. \tag{3.2}$$

Then, we use these values to compute the expected number of times that probes on link k had a delay of i units as

$$M_{k,i}^{(q)} = \sum_{\mathcal{C}_j \in \mathcal{C}: k \in \mathcal{T}^{\mathcal{C}_j}} \sum_{\vec{x} \in \mathcal{X}^{\mathcal{C}_j}: x_k = i} N_{\vec{x}}^{\mathcal{C}_j\ (q)}. \tag{3.3}$$

We also need to keep track of m_k which is the total number of probes that crossed link k.

The M-step is quite simple once the sufficient statistics have been imputed.

$$\alpha_k(i)^{(q)} = \frac{1}{m_k} M_{k,i}^{(q)} \tag{3.4}$$

A detailed example of the E- and M-steps for a simple tree topology is given in a later section. The computationally challenging aspect in our setting is to partition the observed end-to-end delays into the set of possible link delay combinations. These details are given next.

3.2.2 Partitioning

Consider the left panel of Figure 3.1. Suppose that this is the probing tree for a five-cast experiment and that the maximum link delay is $b = 2$. Suppose further that a single probe results in the observed delay vector $Y = [2\ 3\ 3\ 4\ 3]$. We need to systematically partition this end-to-end delay vector into the complete list of all possible link delay vectors that give this result. We use a top-down approach, identifying possible delays for links starting at the top of the tree and moving downward. We begin by listing possible link delays for the first link, between nodes 0 and 1, and leaving the rest of the delays as path delays. This amounts to imagining the tree takes the form of the *shrub* shown in the right panel of Figure 3.1 with each branch of the shrub having a maximum delay determined by b and the number of links from node 1 to each of the receivers.

To get the lower bound of the possible delays for the first link, we consider the minimum delay possible on this link that will give the observed values. The lower bound is the maximum of a set containing zero and each observed value minus the maximum delay that could be obtained on its branch, $Y_r - g_r b$ where g_r is the number of links hidden in the branch of the shrub connecting receiver r to the splitting node. For this example, the value is one. The upper bound is simply the minimum of b

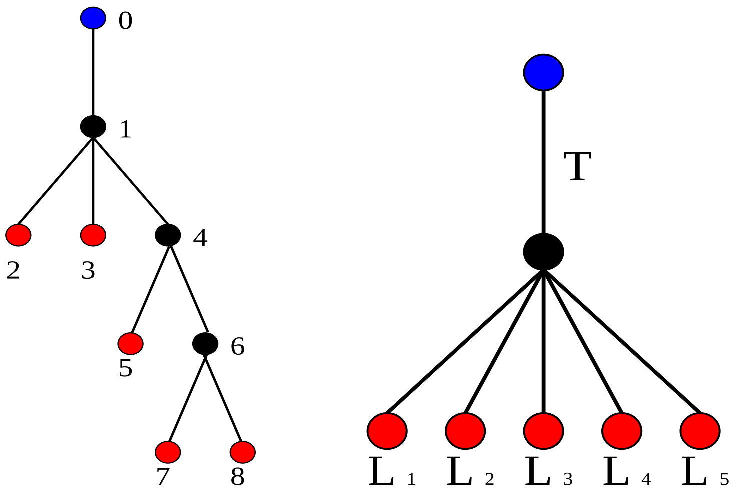

Figure 3.1: Partitioning example: probing subtree (left panel) and its corresponding shrub (right panel).

and the set of observed values. Here the value is two. This allows us to expand the observed delay into the set of link 1 delays and the remaining delays:

$$\begin{bmatrix} 2 & 3 & 3 & 4 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 2 \\ 2 & 0 & 1 & 1 & 2 & 1 \end{bmatrix}. \tag{3.5}$$

We have now isolated the delays that could occur on the first link. We have also isolated the delays that could occur on the second and third links. Now we need to expand the triplets [2 3 2] and [1 2 1]. This is done exactly as before by considering only the portion of the tree rooted on the link between nodes 1 and 4. Each triplet is an end-to-end observation from this portion of the tree. For each set, we imagine the tree to be a three-branch shrub and expand the observation on the possible values that could occur on link 4:

$$\begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 2 \\ 2 & 0 & 1 & 1 & 2 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 2 & 0 & 2 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 2 & 1 \\ 1 & 1 & 2 & 2 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 & 2 & 1 \\ 2 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{3.6}$$

Partitioning the second shrub gave us the range of values for links 4 and 5 leaving all the pairs comprising the last two columns. Each pair can be partitioned into the three component parts of this remaining shrub to give us the full partition for the observed delay on this tree:

$$\begin{bmatrix} 1 & 1 & 2 & 0 & 2 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 2 & 1 \\ 1 & 1 & 2 & 2 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 & 2 & 1 \\ 2 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 2 & 0 & 2 & 1 & 2 & 1 \\ 1 & 1 & 2 & 0 & 2 & 2 & 1 & 0 \\ 1 & 1 & 2 & 1 & 1 & 0 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 2 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 & 0 & 2 & 1 \\ 2 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.7}$$

Formally, let a shrub be any tree graph with single internal node from which has one or more children that are all receivers; see Figure 3.2. Partitioning the shrub is all that is required to partition any tree or subtree. By moving top-down and expanding one link at a time, we can ignore any structure below the link of interest. The tree becomes a shrub by considering each receiver descended from the link of interest to be on a separate branch. After expanding the desired link, the remaining delay can again be partitioned using the shrub algorithm. A single recursive function is all that is required: it would partition the tree as if it were a shrub and then call itself to partition the sub-shrubs.

The algorithm for the general shrub with r receivers is quite simple. Let $Y = [Y_1, \ldots, Y_r]$ be the delay observed on the shrub. Further, let t be the maximum delay that can be observed on the trunk and let l_i be the maximum delay that can be

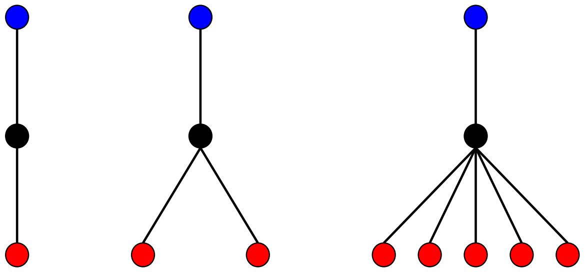

Figure 3.2: Examples of shrubs.

observed on leaf i. We have

$$a = \max\left\{0, \max_i\{Y_i - l_i\}\right\} \text{ and} \tag{3.8}$$

$$z = \min\left\{t, \min_i\{Y_i\}\right\} \tag{3.9}$$

as the minimum and maximum possible values for the trunk delay. Thus, in one for-loop it is easy to make the expansion:

$$Y = [Y_1, \ldots, Y_r] \rightarrow X = \begin{bmatrix} a & Y_1 - a & \ldots & Y_r - a \\ a+1 & Y_1 - (a+1) & \ldots & Y_r - (a+1) \\ \vdots & \vdots & \ddots & \vdots \\ z & Y_1 - z & \ldots & Y_r - z \end{bmatrix}. \tag{3.10}$$

3.2.3 Detailed EM Steps for a Simple Tree

Here we present a concrete example on a simple tree with one bicast and one unicast experiment. Let the collection be $\mathcal{C} = \{\langle 2, 3\rangle, \langle 4\rangle\}$ and let $X_k \in \{0, 1\}$. The observed sufficient statistics are the counts for each outcome of each scheme. Here are the steps for one iteration of the EM algorithm. We start with the E-step.

$$
\begin{aligned}
M_{1,0}^{(q)} &= N_{(0,0)}^{\langle 2,3\rangle} + N_{(0,1)}^{\langle 2,3\rangle} + N_{(1,0)}^{\langle 2,3\rangle} \\
&+ \frac{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1)}{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)} N_{(1,1)}^{\langle 2,3\rangle} \\
&+ N_{(0)}^{\langle 4\rangle} + \frac{\alpha_1^{(q-1)}(0)\alpha_4^{(q-1)}(1)}{\alpha_1^{(q-1)}(0)\alpha_4^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_4^{(q-1)}(0)} N_{(1)}^{\langle 4\rangle} \\
M_{1,1}^{(q)} &= \frac{\alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)}{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)} N_{(1,1)}^{\langle 2,3\rangle} \\
&+ N_{(1,2)}^{\langle 2,3\rangle} + N_{(2,1)}^{\langle 2,3\rangle} + N_{(2,2)}^{\langle 2,3\rangle} \\
&+ N_{(2)}^{\langle 4\rangle} + \frac{\alpha_1^{(q-1)}(1)\alpha_4^{(q-1)}(0)}{\alpha_1^{(q-1)}(0)\alpha_4^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_4^{(q-1)}(0)} N_{(1)}^{\langle 4\rangle} \\
M_{2,0}^{(q)} &= N_{(0,0)}^{\langle 2,3\rangle} + N_{(0,1)}^{\langle 2,3\rangle} + N_{(1,2)}^{\langle 2,3\rangle} \\
&+ \frac{\alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)}{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)} N_{(1,1)}^{\langle 2,3\rangle} \\
M_{2,1}^{(q)} &= N_{(1,0)}^{\langle 2,3\rangle} + N_{(2,1)}^{\langle 2,3\rangle} + N_{(2,2)}^{\langle 2,3\rangle} \\
&+ \frac{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1)}{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)} N_{(1,1)}^{\langle 2,3\rangle} \\
M_{3,0}^{(q)} &= N_{(0,0)}^{\langle 2,3\rangle} + N_{(1,0)}^{\langle 2,3\rangle} + N_{(2,1)}^{\langle 2,3\rangle} \\
&+ \frac{\alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)}{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)} N_{(1,1)}^{\langle 2,3\rangle} \\
M_{3,1}^{(q)} &= N_{(0,1)}^{\langle 2,3\rangle} + N_{(1,2)}^{\langle 2,3\rangle} + N_{(2,2)}^{\langle 2,3\rangle} \\
&+ \frac{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1)}{\alpha_1^{(q-1)}(0)\alpha_2^{(q-1)}(1)\alpha_3^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_2^{(q-1)}(0)\alpha_3^{(q-1)}(0)} N_{(1,1)}^{\langle 2,3\rangle} \\
M_{4,0}^{(q)} &= N_{(0)}^{\langle 4\rangle} + \frac{\alpha_1^{(q-1)}(1)\alpha_4^{(q-1)}(0)}{\alpha_1^{(q-1)}(0)\alpha_4^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_4^{(q-1)}(0)} N_{(1)}^{\langle 4\rangle} \\
M_{4,1}^{(q)} &= \frac{\alpha_1^{(q-1)}(0)\alpha_4^{(q-1)}(1)}{\alpha_1^{(q-1)}(0)\alpha_4^{(q-1)}(1) + \alpha_1^{(q-1)}(1)\alpha_4^{(q-1)}(0)} N_{(1)}^{\langle 4\rangle} + N_{(2)}^{\langle 4\rangle}
\end{aligned}
$$

Here is the M-step.

$$\begin{aligned}
\alpha_1^{(q)}(0) &= M_{1,0}^{(q)}/(n^{\langle 2,3\rangle}+n^{\langle 4\rangle}) \\
\alpha_1^{(q)}(1) &= M_{1,1}^{(q)}/(n^{\langle 2,3\rangle}+n^{\langle 4\rangle}) \\
\alpha_2^{(q)}(0) &= M_{2,0}^{(q)}/n^{\langle 2,3\rangle} \\
\alpha_2^{(q)}(1) &= M_{2,1}^{(q)}/n^{\langle 2,3\rangle} \\
\alpha_3^{(q)}(0) &= M_{3,0}^{(q)}/n^{\langle 2,3\rangle} \\
\alpha_3^{(q)}(1) &= M_{3,1}^{(q)}/n^{\langle 2,3\rangle} \\
\alpha_4^{(q)}(0) &= M_{4,0}^{(q)}/n^{\langle 4\rangle} \\
\alpha_4^{(q)}(1) &= M_{4,1}^{(q)}/n^{\langle 4\rangle}
\end{aligned}$$

3.2.4 EM Complexity

To study the complexity of an EM iteration, consider first a specific k-cast scheme. There are $b^{|\mathcal{T}^{\mathcal{C}_j}|}$ link delay outcomes for this probe. For each of these, there are $|\mathcal{T}^{\mathcal{C}_j}|$ multiplications to compute the probability of the link delay outcome. For each outcome, there is also a single addition to tally up the end-to-end probabilities and a single division to compute the conditional probability of each outcome given the end-to-end outcome. Finally, there are $|\mathcal{T}^{\mathcal{C}_j}|$ additions to tally up the sufficient statistics. Overall, this gives us $\mathcal{O}\{b^{|\mathcal{T}^{\mathcal{C}_j}|}\}$ operations. The largest subtree sets the complexity for the E-step at $\mathcal{O}\{b^{|\mathcal{T}^{\mathcal{C}_{large}}|}\}$ where $\mathcal{C}_{large}$ is the largest experiment. The M-step is trivial consisting of $|\mathcal{E}b|$ divisions.

There are a few things to note. First, mixtures of bicast and unicast schemes offer the best complexity while meeting the identifiability conditions. Additionally, they scale better than k-casts with larger values of k. In particular, an omnicast scheme does not scale well.

If the tree grows in size but not in depth, then the bicast schemes should scale well because this will simply result in more bicast schemes rather than more complicated bicast schemes. This property does not hold for omnicast schemes which have complexity $\mathcal{O}\{b^{|\mathcal{E}|}\}$.

Unfortunately, the EM-algorithm does not scale well as the tree gets deeper for any k-cast scheme. For such topologies, we explore alternative fast estimators in a later section.

Also of note, the complexity stated here is the extreme worst case based on observing every possible delay combination from every probing experiment. In practice, both the partitioning and estimation only consider the observed delays which will significantly reduce the average-case complexity.

Remark 4: The EM algorithm can be made computationally more efficient through parallelization. Notice that the E-step involves computing a sum that ranges first over the schemes and then over the outcomes for that experiment. This sum can be broken down into component pieces which can be computed simultaneously and combined. Here is one possible implementation of this idea: Suppose that the data from each scheme in an experiment of size $|\mathcal{C}|$ are collected and processed by an independent processor. In addition, another processor handles the collection and distribution of information among the remaining $|\mathcal{C}|$ processors. Specifically, its duties include the collection of sufficient statistics, performing the M-step and distributing back the estimated parameters to the other processors. Each processor in charge of a scheme requires knowledge of the parameters associated with its scheme's subtree and its data. Their responsibility in calculating the E-step is to compute their corresponding part of the sum.

3.2.5 Numerical Investigation of EM-algorithm

This section studies effects of the tree size and the number of bins (in the discrete delay distribution) on the convergence of the EM algorithm. These two factors determine the number of model parameters. In practice, one can choose the number of bins but has limited control over the tree size. For example, in a monitoring situation, a coarse distribution can be estimated quickly and still provide enough information to detect anomalies in the network. On the other hand, one can only obtain a smaller tree by lumping several links together and eliminating some of the receivers.

Consider first the effect of the tree size, in terms of both number of links and layers. We start with a two-layer symmetric binary tree so that there is a total of three links. Then, we add two children at a time. So the five-link tree corresponds to adding two children to one of the receiver nodes. The seven-link tree adds two children to the other receiver nodes (so that this is just a three-layer symmetric binary tree). We proceed in this manner until we get the four-layer symmetric binary tree with 15 links (see x-axis of left panel of Figure 3.3). The remaining model components are held fixed. In particular, each link has a five-bin uniform delay distribution chosen because it provides maximum entropy, thus it is the most difficult to resolve. We use a flexicast experiment that is a minimal set of bicast schemes that satisfy the identifiability conditions. Some additional studies indicated that convergence for the EM algorithm does not seem to depend greatly on the number of probes used. Hence, in the investigations here, each bicast scheme had 1000 probes for each links in its subtree. For each tree size, 50 sets of data were generated and used for estimation. The convergence criteria was a change in the log-likelihood of less than 10^{-4}. The left panel of Figure 3.3 shows the number of iterations required for convergence for each data set. The average number of iterations for each size is plotted with standard deviation error bars. This suggests that the average number of iterations seems to be increasing at a faster than linear rate (perhaps exponential) with the number of links. This is a further indication that the EM algorithm does not scale well to larger networks.

Next, consider the effect of the number of bins in each link with uniform delay distributions. The number of bins on each link was varied from two to 15. We considered both two-layer and three-layer binary symmetric trees with three and seven links respectively. Again, a minimal bicast experiment with 1000 probes per scheme was used. The results for both trees are shown in the right panel of Figure 3.3. In this scenario, the average number of iterations seems to grow approximately linearly with the number of bins on each link. This is an important observation that we will exploit later in developing a faster algorithm.

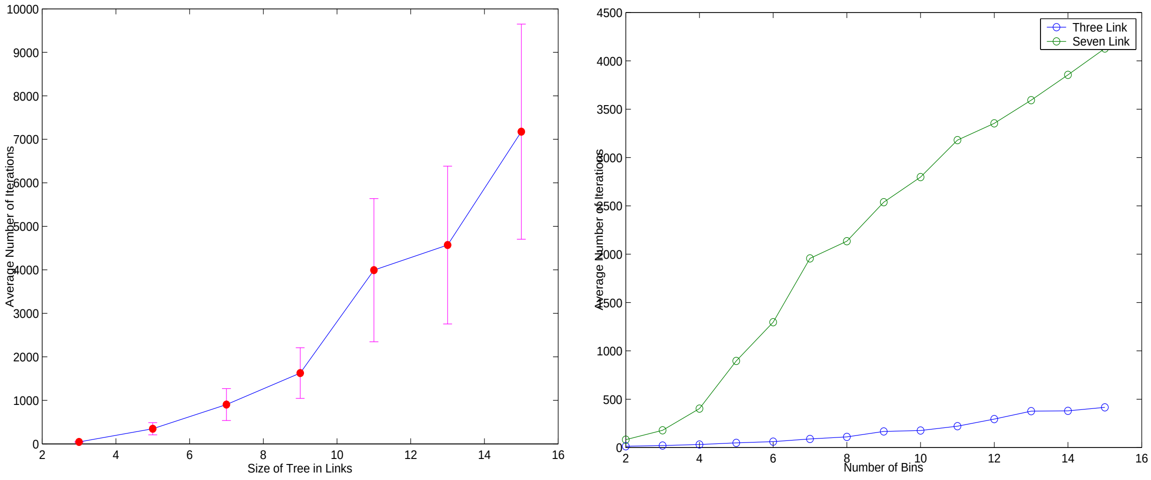

Figure 3.3: Left panel: Number of iterations versus tree size (number of links); Right panel: Number of iterations versus bin size for 2- and 3-layer trees

3.2.6 Asymptotic Properties of the MLE

Given the multinomial nature of the underlying k-cast schemes, the asymptotic properties of the MLE mostly follow from general principles. The difference arises because the flexicast experiments mean that individual probes are not *i.i.d.* The proposition below establishes that Fisher information matrix is positive definite at the true value $\vec{\alpha}_0$. Thus, the likelihood has a unique maximum in a local neighborhood of the true value $\vec{\alpha}_0$.

Proposition III.2. *The Fisher information matrix $\mathcal{I}(\vec{\alpha}_0)$ for the maximum likelihood estimator based on end-to-end quantized measurements from a flexicast experiment $\mathcal{C}$ is finite and positive definite.*

Proof: Here we prove the special case that the Fisher information matrix from an identifiable collection of bicast and unicast schemes is finite and positive definite. This proof is given in order to simplify the presentation. It is easy to extend this proof to a general experiment by considering the bicast projections of more complex experiments.

First, it is easy to see that the Fisher information is finite everywhere for $\vec{\alpha}$ such that $\alpha_k(i) \in (0,1)$ for all k and i. Consider the score functions:

$$S_{k,i}(\vec{\alpha}) = \frac{\partial l(\vec{\alpha}; \mathbf{Y})}{\partial \alpha_k(i)}. \tag{3.11}$$

Form the vector of score functions $\vec{S}$ by stacking the individual functions in the same order as $\vec{\alpha}$. Because the log-likelihood has continuous partial derivatives, we know that $\mathrm{E}\{S_{k,i}\} = 0$ for k and i. Further, the Fisher information is the covariance matrix of $\vec{S}$. As a result, the Fisher $\mathcal{I}$ is positive, semi-definite. We will show that it is actually positive definite.

Suppose that $\vec{c}'\mathcal{I}\vec{c} = 0$ for some $\vec{\alpha}$. As this last quantity is the variance of a random variable with zero expectation, $\vec{c}'\vec{S}$, we have $\vec{c}'\vec{S} = 0$, almost surely, or

$$\sum_{k=1}^{|\mathcal{E}|}\sum_{i=0}^{b-1} c_{k,i}\frac{\partial l(\vec{\alpha};\mathbf{Y})}{\partial \alpha_k(i)} = 0, \tag{3.12}$$

for all possible $\mathbf{Y}$. We will consider various experimental outcomes and show that the above implies that $\vec{c} = \vec{0}$, thus $\mathcal{I}$ is non-singular and positive definite.

First note that all links except those connecting to receivers must be covered by at least one bicast pair. All receiver links have the possibility of being covered by just unicast schemes.

Consider some pair $\langle k, l\rangle$ that splits at node 1. Let every scheme including this one receive one probe that experiences no end-to-end delay at any endpoint. This gives us the following quantity of interest.

$$\vec{c}'\vec{S} \;=\; \frac{c_{1,0}}{\alpha_1(0)} + \sum_{i\in\mathcal{P}_{1,k}} \frac{c_{i,0}}{\alpha_i(0)} + \sum_{i\in\mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0.$$

The term $g(\vec{c})$ covers the information coming from the other schemes. Rewrite the terms concerning the k branch using the path probability and the chain rule.

$$\begin{aligned}
\vec{c}'\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} + \sum_{i\in\mathcal{P}_{1,k}} \frac{c_{i,0}}{\pi_{1,k}(0)}\frac{\partial \pi_{1,k}(0)}{\partial \alpha_i(0)} + \sum_{i\in\mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0\\
\vec{c}'\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} + \frac{1}{\pi_{1,k}(0)}\sum_{i\in\mathcal{P}_{1,k}} c_{i,0}\frac{\partial \pi_{1,k}(0)}{\partial \alpha_i(0)} + \sum_{i\in\mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0\\
\vec{c}'\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} + \frac{1}{\pi_{1,k}(0)} w_{1,k,0} + \sum_{i\in\mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0
\end{aligned}$$

Let each scheme again receive one probe, but let the probe sent to $\langle k, l\rangle$ experience delay pair $(d, 0)$ for some $d < |\mathcal{P}_{1,k}|b$ while the other schemes still result in no delay.

Again, write the quantity using the path probabilities and the chain rule.

$$\begin{aligned}\vec{c}\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} + \frac{1}{\pi_{1,k}(d)} \sum_{i \in \mathcal{P}_{1,k}} \sum_{j=0}^{d \wedge (b-1)} c_{i,j} \frac{\partial \pi_{1,k}(d)}{\partial \alpha_i(j)} + \sum_{i \in \mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0 \\ \vec{c}\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} + \frac{1}{\pi_{1,k}(d)} w_{1,k,d} + \sum_{i \in \mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0\end{aligned}$$

Subtracting this equation from the equation resulting from the $(0,0)$ outcome, it is clear that

$$w_{1,k,d} = \frac{\pi_{1,k}(d)}{\pi_{1,k}(0)} w_{1,k,0}.$$

Let $M = |\mathcal{P}_{1,k}|b$, the maximum delay that can occur on the k branch. Consider the $(M,0)$ outcome. Note that the path probability $\pi_{1,k}(M)$ is not a free parameter since the probabilities must sum to one. We get the following equation.

$$\begin{aligned}\vec{c}\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} + \frac{1}{\pi_{1,k}(M)} \sum_{d=0}^{M-1} \frac{\partial \pi_{1,k}(M)}{\partial \pi_{1,k}(d)} w_{1,k,d} + \sum_{i \in \mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0 \\ \vec{c}\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} - \frac{1}{\pi_{1,k}(M)} \sum_{d=0}^{M-1} w_{1,k,d} + \sum_{i \in \mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0 \\ \vec{c}\vec{S} &= \frac{c_{1,0}}{\alpha_1(0)} - \frac{1}{\pi_{1,k}(M)} \sum_{d=0}^{M-1} \frac{\pi_{1,k}(d)}{\pi_{1,k}(0)} w_{1,k,0} + \sum_{i \in \mathcal{P}_{1,l}} \frac{c_{i,0}}{\alpha_i(0)} + g(\vec{c}) = 0\end{aligned}$$

Subtracting this equation from the $(0,0)$ equation we see that $w_{1,k,0} = 0$ and thus $w_{1,k,d} = 0$ for all d. The same argument gives us $w_{1,l,d} = 0$ for all d.

Now we focus on the trunk. Note that our new equation for the $(0,0)$ outcome is

$$\vec{c}\vec{S} = \frac{c_{1,0}}{\alpha_1(0)} + g(\vec{c}) = 0.$$

Some additional notation is required to simplify the mathematics. Let $\gamma_{\langle k,l \rangle}(i,j)$ be the end-to-end pair probability. Let $\Delta^s_{\langle k,l \rangle}(x, y_k, y_l)$ be the probability that $X_k = x$ and $(Y_k, Y_l) = (y_k, y_l)$.

Consider the $(1,1)$ outcome for this scheme while the other schemes still have no delays. We get the following.

$$\begin{aligned}
\vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(1,1)}\left\{c_{1,0}\frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\alpha_1 0} + c_{1,1}\frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\alpha_1 1}\right. \\
&+ \frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\pi_{1,k}(0)}w_{1,k,0} + \frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\pi_{1,k}(1)}w_{1,k,1} \\
&+ \left.\frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\pi_{1,l}(0)}w_{1,l,0} + \frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\pi_{1,l}(1)}w_{1,l,1}\right\} + g(\vec{c}) = 0 \\
\vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(1,1)}\left\{c_{1,0}\frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\alpha_1 0} + c_{1,1}\frac{\partial\gamma_{\langle k,l\rangle}(1,1)}{\partial\alpha_1 1}\right\} + g(\vec{c}) = 0 \\
\vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(1,1)}\left\{c_{1,0}\frac{\Delta^1_{\langle k,l\rangle}(0,1,1)}{\alpha_1(0)} + c_{1,1}\frac{\Delta^1_{\langle k,l\rangle}(1,0,0)}{\alpha_1(1)}\right\} + g(\vec{c}) = 0
\end{aligned}$$

Subtracting this equation from the $(0,0)$ equation, we see the following.

$$\begin{aligned}
\frac{c_{1,0}}{\alpha_1(0)}\left[1 - \frac{\Delta^1_{\langle k,l\rangle}(0,1,1)}{\gamma_{\langle k,l\rangle}(1,1)}\right] - \frac{c_{1,1}}{\alpha_1(1)}\frac{\Delta^1_{\langle k,l\rangle}(1,0,0)}{\gamma_{\langle k,l\rangle}(1,1)} &= 0 \\
\frac{c_{1,0}}{\alpha_1(0)}\frac{\Delta^1_{\langle k,l\rangle}(1,0,0)}{\gamma_{\langle k,l\rangle}(1,1)} - \frac{c_{1,1}}{\alpha_1(1)}\frac{\Delta^1_{\langle k,l\rangle}(1,0,0)}{\gamma_{\langle k,l\rangle}(1,1)} &= 0 \\
c_{1,0}\frac{\alpha_1(1)}{\alpha_1(0)} &= c_{1,1}
\end{aligned}$$

Proceed by induction. Consider the (d,d) outcome for $d < b$ and assume that

$$c_{1,i} = c_{1,0}\frac{\alpha_1(i)}{\alpha_1(0)},$$

for all $i < d$. We have the following equation for this outcome.

$$\begin{aligned}
\vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(d,d)}\sum_{i=0}^{d} c_{1,i}\frac{\Delta^1_{\langle k,l\rangle}(i,d-i,d-i)}{\alpha_1(i)} + g(\vec{c}) = 0 \\
\vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(d,d)}\left[\sum_{i=0}^{d-1} c_{1,0}\frac{\alpha_1(i)}{\alpha_1(0)}\frac{\Delta^1_{\langle k,l\rangle}(i,d-i,d-i)}{\alpha_1(i)} + c_{1,d}\frac{\Delta^1_{\langle k,l\rangle}(d,0,0)}{\alpha_1(d)}\right] + g(\vec{c}) = 0
\end{aligned}$$

Subtracting this equation from the $(0,0)$ equation we once again arrive at the relationship

$$c_{1,d} = c_{1,0}\frac{\alpha_1(d)}{\alpha_1(0)}.$$

Keeping in mind that $\alpha_1(b)$ is not a free parameter, here is the equation for the (b,b) outcome.

$$\begin{aligned}\vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(1,1)}\sum_{i=0}^{b-1}\frac{c_{1,i}}{\alpha_1(i)}\left[\Delta^1_{\langle k,l\rangle}(i,b-i,b-i)-\Delta^1_{\langle k,l\rangle}(i,0,0)\right]+g(\vec{c})=0\\ \vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(1,1)}\sum_{i=0}^{b-1}\frac{c_{1,0}}{\alpha_1(0)}\left[\Delta^1_{\langle k,l\rangle}(i,b-i,b-i)-\Delta^1_{\langle k,l\rangle}(i,0,0)\right]+g(\vec{c})=0\end{aligned}$$

Finally, subtracting this equation from the $(0,0)$ equation, we can see that $c_{1,0}=0$ and thus $c_{1,d}=0$ for all d.

Again, we proceed by induction. Take some internal node s and let $\langle k,l\rangle$ be a pair that splits at this node. Assume that $c_{i,j}=0$ for all $i\in\mathcal{P}_{0,s}$ except for s and all $j\in\{0,1,\ldots,(b-1)\}$. Use the argument from above to show that $w_{s,k,d}=0$ and $w_{s,l,d}=0$ for all d.

For the $(0,0)$ outcome, we simply have

$$\vec{c}\vec{S} = \frac{c_{s,0}}{\alpha_s(0)}+g(\vec{c})=0.$$

Consider the (d,d) equation where $d<b$ and assume that

$$c_{s,i}=c_{s,0}\frac{\alpha_s(i)}{\alpha_s(0)},$$

for all $i<d$. Note that this is clearly true for $i=0$.

$$\begin{aligned}\vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(d,d)}\sum_{i=0}^{d}c_{s,i}\frac{\partial\gamma_{\langle k,l\rangle}(d,d)}{\partial\alpha_s(i)}+g(\vec{c})=0\\ \vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(d,d)}\sum_{i=0}^{d}c_{s,i}\frac{\Delta^s_{\langle k,l\rangle}(i,d-i,d-i)}{\alpha_s(i)}+g(\vec{c})=0\\ \vec{c}\vec{S} &= \frac{1}{\gamma_{\langle k,l\rangle}(d,d)}\left[\frac{c_{s,d}}{\alpha_s(d)}\Delta^s_{\langle k,l\rangle}(d,0,0)+\sum_{i=0}^{d-1}\frac{c_{s,0}}{\alpha_s(0)}\Delta^s_{\langle k,l\rangle}(i,d-i,d-i)\right]+g(\vec{c})=0\end{aligned}$$

Subtracting this equation from the $(0,0)$ equation, we confirm the relationship

$$c_{s,d}=c_{s,0}\frac{\alpha_s(d)}{\alpha_s(0)}.$$

Finally, consider the (b, b) outcome.

$$\vec{c}\vec{S} = \frac{1}{\gamma_{\langle k,l\rangle}(b,b)} \left\{ \sum_{d=0}^{b-1} \frac{c_{s,d}}{\alpha_s(d)} [\Delta^s_{\langle k,l\rangle}(d, b-d, b-d) - \Delta^s_{\langle k,l\rangle}(d, 0, 0)] \right\} + g(\vec{c}) = 0$$

$$\vec{c}\vec{S} = \frac{1}{\gamma_{\langle k,l\rangle}(b,b)} \frac{c_{s,0}}{\alpha_s(0)} \left\{ \sum_{d=0}^{b-1} [\Delta^s_{\langle k,l\rangle}(d, b-d, b-d) - \Delta^s_{\langle k,l\rangle}(d, 0, 0)] \right\} + g(\vec{c}) = 0$$

Again, we subtract this from the $(0, 0)$ equation and can see that $c_{s,d} = 0$ for all d.

The above argument covers all links except for receiver links. For receiver links that are covered by a bicast pair, the argument used above on the branches will suffice. We focus now on receivers covered by only unicast schemes. Consider an scheme for receiver k. Note that $c_{i,j} = 0$ for all links above k. Consider an outcome in which all schemes receive one probe and they all experience no delay. We have the following quantity of interest.

$$\vec{c}\vec{S} = \frac{c_{k,0}}{\alpha_k(0)} + g(\vec{c}) = 0$$

If we change the outcome of this scheme to have delay 1, we get

$$\vec{c}\vec{S} = \frac{c_{k,0}}{\alpha_k(0)} \frac{\alpha_k(0)\pi_{0,f(k)}(1)}{\pi_{0,k}(1)} + \frac{c_{k,1}}{\alpha_k(1)} \frac{\alpha_k(1)\pi_{0,f(k)}(0)}{\pi_{0,k}(1)}$$

From these two equations, we see the usual relationship. By induction, we can show that it hold in general:

$$c_{k,d} = c_{k,0} \frac{\alpha_k(d)}{\alpha_k(0)}.$$

Next, we examine the equation for a delay of b.

$$\vec{c}\vec{S} = \sum_{i=0}^{b-1} \frac{c_{k,i}}{\alpha_k(i)} \left[\alpha_k(i)\pi_{0,f(k)}(b-i) - \alpha_k(i)\pi_{0,f(k)}(0])\right] + g(\vec{c}) = 0$$

$$\vec{c}\vec{S} = \sum_{i=0}^{b-1} \frac{c_{k,0}}{\alpha_k(0)} \left[\alpha_k(i)\pi_{0,f(k)}(b-i) - \alpha_k(i)\pi_{0,f(k)}(0])\right] + g(\vec{c}) = 0$$

From this and the 0 equation, we can see that $c_{k,d} = 0$ for all d and the proof is finished. □

Proposition III.3. *Let* $\frac{n^{\mathcal{C}_j}}{n} \to \lambda^{\mathcal{C}_j}$ *as* $n \to \infty$ *with* $0 < \lambda^{\mathcal{C}_j} < 1$ *for* $j = 1, ..., M$. *Then,*

$$\vec{\alpha}_{\text{MLE}} \to \vec{\alpha}_0, a.s, \tag{3.13}$$

and is asymptotically normal and fully efficient; i.e.,

$$\sqrt{\mathbf{n}}(\vec{\alpha}_{\text{MLE}} - \vec{\alpha}_0) \Rightarrow Z, \tag{3.14}$$

where $Z \sim N\{\vec{0}, \mathcal{I}^{-1}(\vec{\alpha})\}$.

Proof: Here we prove consistency. Again, we present the special case in which $\mathcal{C}$ is a collection of bicast and unicast schemes. This is illustrative of the main issues in the proof and is conceptually easy to extend.

Let $\vec{\gamma}^0$ denote the true end-to-end probabilities for a particular collection of schemes. Note that the end-to-end outcomes of the schemes are multinomial and that the simple method of moments estimators are the MLEs and are strongly consistent, *i.e.*

$$\hat{\gamma}_c(y) = \frac{N_y^c}{n^c} \to \gamma_c^0(y), \text{ as } n^c \to \infty. \tag{3.15}$$

Rewrite the likelihood in terms of these probabilities and set the score functions equal to zero:

$$\frac{\partial l(\vec{\alpha}; \hat{\vec{\gamma}})}{\partial \alpha_k(i)} = \sum_{c \in \mathcal{C}} \sum_{\vec{y} \in \mathcal{Y}^c} n^c \frac{\hat{\gamma}_c(\vec{y})}{\gamma_c(\vec{y})} \frac{\partial \gamma_c(\vec{y})}{\partial \alpha_k(i)} = 0. \tag{3.16}$$

Since the expected value of the score functions is zero, one solution to this set of equation is $(\vec{\alpha}^0, \vec{\gamma}^0)$. Because the Fisher information matrix is positive definite, we can apply the Implicit Function Theorem. Thus, there exists an open set $\mathcal{G}$ containing $\vec{\gamma}^0$ and an open set $\mathcal{A}$ containing $\vec{\alpha}^0$ with the property that for each $\vec{\gamma} \in \mathcal{G}$ there is a unique $\vec{\alpha}(\vec{\gamma}) \in \mathcal{A}$ such the pair solves the score functions. Further, the function $\vec{\alpha}(\vec{\gamma})$

is differentiable. Because $\hat{\vec{\gamma}}$ is strongly consistent and $\vec{\alpha}(\vec{\gamma})$ is continuous in a region around the true parameter $\vec{\gamma}^0$, we get the result. □

The normality and efficiency of $\vec{\alpha}$ can be obtained similarly.

3.3 Faster, Heuristic Algorithms

As noted before, the full EM algorithm does not scale well to large-scale networks. Thus, we consider alternative techniques that are faster. A new algorithm is proposed below and compared with two other methods that have been proposed in the literature.

3.3.1 Grafting

We propose a method called grafting which computes the "local MLE" on each subtree and uses peeling to combine the results. In essence, this treats each k-cast scheme as an omnicast experiment on the probing subtree. It uses EM to solve for the MLE of the logical links on this subtree and then peel to get estimates for individual links. For collections of bicast and unicast scheme, this technique scales very well because the EM algorithm is applied to a series of three-link, two-layer trees. Thus the complexity is a cubic polynomial in the number of bins, a vast improvement over the standard MLE complexity. Based on the investigations discussed previously, increasing the number of bins on the links increases the average iterations approximately linearly while adding links increases the required iterations exponentially. This local scheme takes advantage of this fact by trading links for bins.

We will explain the details using just a flexicast experiment with bicast and unicast scheme. First, consider a bicast scheme and the corresponding subtree. Let the trunk have t links and the two branches have l_1 and l_2 links respectively. The subtree has just three logical links with varying numbers of bins on each: the trunk has $tb+1$ bins and the branches have l_1b+1 and l_2b+1 bins. We apply the EM algorithm to this logical subtree and solve for its MLE. This is done for all of the bicast schemes. This gives the estimates for the trunks and branches of all the bicast subtrees.

Individual links can be now be obtained in one of several ways. Consider first top-down peeling that was discussed earlier. This is simple and non-iterative although not as efficient as another scheme to be discussed shortly. At least one pair must split at node 1, so at least one of the local MLEs must give us an estimate for link 1. Now, at least one scheme gives us the local MLE for the path from the root node to every child of node 1. So the individual links up to these points can be identified through peeling. This process continues down the tree identifying each link. The receivers covered by bicast experiments can be identified as the branches in a subtree or by peeling from the branches. The receivers covered by only unicast experiments can also be identified by peeling.

We actually propose a more sophisticated peeling mechanism. This is a fixed-point type algorithm that rises from postulating an EM algorithm for imaginary data. Imagine that we send n probes across the path. Form data by setting $n_d = n\pi_{0,2}(d)$. The data are counts of the number of times delay d was observed on the path for all possible d. In the E step, we want to compute M_i, the expected number of times that delay i was seen on the unknown link. After the q-th iteration, this is given by:

$$M_i^{(q+1)} = \sum_{j=0}^{b} \frac{\alpha_2^{(q)}(i)\alpha_1(j)}{\pi_{0,2}^{(q)}(i+j)} n_{i+j}, \tag{3.17}$$

where $\vec{\pi}_{0,2}^{(q)}$ is updated with each update of $\vec{\alpha}_2^{(q)}$. Note that this is not the quantity used to generate the data. Since we obtain our estimates by dividing M_i by n, we get the following equation:

$$\alpha_2(i)^{(q+1)} = \sum_{j=0}^{b} \frac{\alpha_2^{(q)}(i)\alpha_1(j)}{\pi_{0,2}^{(q)}(i+j)} \pi_{0,2}(i+j). \tag{3.18}$$

This equation is no longer based on our imaginary data and can be implemented. It is run until $\vec{\alpha}_2$ approaches its fixed point. Note that this fixed-point algorithm meets standard conditions for convergence; see Istratescu (1981). Unlike the top-down method, this peeling function uses all of the information from the two known distributions.

The peeling method can lead to multiple estimates for some links. The easiest way to address this problem is is to combine them, using either a simple average or a weighted average. For the latter, if we have two estimates of $\vec{\alpha}_1$ from experiments $\mathcal{C}_1$ and $\mathcal{C}_2$, we can combine them as follows to get

$$\tilde{\vec{\alpha}}_1 = \frac{n^{\mathcal{C}_1}\vec{\alpha}^{\mathcal{C}_1} + n^{\mathcal{C}_2}\vec{\alpha}^{\mathcal{C}_2}}{n^{\mathcal{C}_1} + n^{\mathcal{C}_2}}. \tag{3.19}$$

It can be shown that the grafting algorithm yields estimators that are consistent and asymptotically normal. The computation of the asymptotic variance is involved. The simplest way to compute the variance is to use bootstrap or other resampling methods.

3.3.2 Other Estimators in the Literature

For the delay tomography problem, two other estimating procedures have been proposed in the literature, both based on omnicast probing. Both use similar modeling assumptions to those presented here: discrete delay with temporal and spatial independence. We will describe them briefly and then compare them to the grafting estimator presented above.

The first, discussed by Lo Presti et al. (2002), depends on solving polynomials. At some link k, the estimator uses the data from the subtree rooted at the link to create a polynomial for each unit of delay, i. The degree of the polynomial is $|\mathcal{D}(k)| - 1$. The second root of this polynomial give us the cumulative probability of delay i on link k. The principal drawback of this estimator is that it does not use all of the information available. End-to-end delays that are larger than the largest allowable link delay are ignored. Additionally, the nature of the estimator allows inappropriate results from the polynomial solution such as negative values or values greater than one.

The estimator by Liang and Yu (2003) is based on a pseudo-likelihood approach. The complexity of the omnicast experiment is reduced by looking at just bicast projections – all pairwise combinations – and using a pseudo-likelihood that treats them as independent. For example, the network in Figure 2.2 has 11 receivers, so the omnicast experiment result in 11-dimensional delay observations. There are 55 possible pairs of receivers, so the pseudo-likelihood scheme treats all the possible pairs of delays as 55 independent bicast observations. The motivating idea is that processing the data as pairs is computationally much more efficient than processing the omnicast data. This can be justified if the gain in computational speed offsets the loss in statistical efficiency.

3.3.3 A Comparison of the Various Estimators

Computational Efficiency

Computational speed of the estimators is an important consideration in real applications. For instance, network monitoring requires the ability to solve the inverse estimation problem very quickly. Here, we investigate the computational efficiencies of the various methods for several different tree structures.

Three-Layer, Binary, Symmetric Tree:

We compare the efficiencies of the MLEs based on omnicast probing, all-pairs

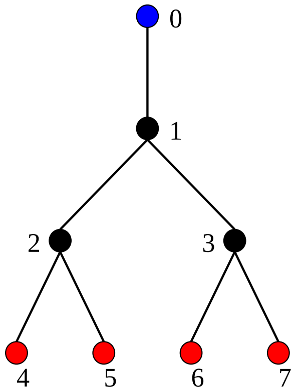

Figure 3.4: Three-layer, binary, symmetric tree.

bicast experiment, and *min+1* bicast experiment. Recall that a minimal flexicast refers to a combination of bicast and unicast probing schemes that satisfy the identifiability condition. For a symmetric binary tree, this consists of just bicasts. To see this, consider the three-layer, binary, symmetric tree in Figure 3.4. The minimal bicast experiment is $\{\langle 4,5\rangle, \langle 5,6\rangle, \langle 6,7\rangle\}$. This experiment is unbalanced as receiver links 4 and 7 are only covered once while links 5 and 6 are covered twice. A more balanced approach is obtained by adding pair $\langle 4,7\rangle$. This ensures that each link on a particular layer of a binary, symmetric tree is covered by the same number of experiments. We refer to such experiments as *min+1* flexicast experiments.

In addition to the MLEs, we also consider the pseudo-likelihood estimator, Lo Presti et al.'s polynomial estimator, and grafting for all-pairs bicast and *min+1* bicast. All of the estimators were implemented using Matlab with the combinatorial partitioning components of the likelihood-based methods written in C. The link delays follow a five-bin truncated geometric distribution. The parameter of the distributions were varied in a manner to keep the situations realistic: the interior links have high probability of zero as compared with edge links to simulate the difference between internal links with large bandwidth and smaller local links. The efficiency comparisons were based on simulated 100 data sets and are shown in Table 3.1.

Estimator	Time
MLE	46.93s
PLE	32.68s
Polynomial	3.96s
All Pairs MLE	17.19s
Min+1 MLE	11.11s
All Pairs Graft	9.11s
Min+1 Graft	5.59s

Table 3.1: Three-Layer Tree Comparison: Average computing time (in seconds).

The polynomial estimator is, of course, the fastest. This is partially driven by the fact that it is solving linear equations in this example (binary tree) and the formulas for the estimates are obtained explicitly. The effect of having a large number of children on the polynomial estimator will be investigated later. We will also see later that this algorithm can be considerably inefficient in a statistical sense. As to be expected, the PLE is faster than the MLE based on the full EM; however, it does not gain as much over the MLE as does the pure bicast algorithm. The all-pairs bicast is more than twice as fast as the likelihood-based multicast estimators while the PLE does not seem to benefit from an order of magnitude gain.

Unbalanced Tree: Here we consider the tree structure shown in Figure 2.2. We investigate only the pseudo-likelihood estimator, the minimum-pairs bicast MLE, the all-pairs grafting procedure, and the minimum pairs grafting procedure. The polynomial estimator was dropped due to the difficulty in implementing it for general trees. It is studied below however for another situation. The link delay distributions were again chosen to be a five-bin, truncated geometric. Table 3.2 shows the results of the comparisons. The pseudo-likelihood is the slowest. The all-pairs grafting requires a tenth of the computation time of the PLE despite sharing similar amounts of data. The flexicast experiment with minimum bicasts has a smaller number of pairs, so it should be expected to save in computational time. The full MLE for this minimum

Estimator	PLE	Min Pairs MLE	All Pairs Graft	Min Pairs Graft
Avg. Time	401.27s	10.12s	41.61s	4.99s

Table 3.2: Computational speeds of estimators applied to data from Figure 2.2

Children	2	6	10
Grafting	.48s	.84s	.99s
Polynomial	1.02s	1.08s	1.13s

Table 3.3: Comparison of computation time for Grafting and the Polynomial estimator on shrubs with varying numbers of children.

pairs experiment is about 40 times faster than the PLE in this example. The grafting procedure with minimum pairs is extremely fast, comparable in speed to the time achieved by the polynomial estimator on the simpler tree discussed previously.

Shrub Comparison: Here we investigate only the two fastest estimators: the grafting procedure with minimum bicast pairs and the polynomial estimator. We study at a set of simple cases: shrubs with increasing numbers of children to see how the performance varies. For each configuration, we generated 1000 data sets from truncated geometric distributions on each link. Table 3.3 lists the average computation times for shrubs with two, size, and 10 children. The grafting procedure is uniformly faster on this test, even when it has to combine information from five trees in the 10-child example. The polynomial estimator performs at its best on small trees with small numbers of bins. However, when the true bin probability is small, we found that it can lead to negative estimates in a significant number of cases.

Statistical Efficiency

Statistical efficiency has received little attention in the literature, perhaps because of the inherent assumption that a large number of probes can be generated easily. In reality, however, active experimentation perturbs the network, and so too much probing in a short period of time can end up causing delay and losses on the network. If we spread the probing over extended period, it will invalidate the stationarity assumption. As a result, one has to limit the number of probes, so any effective estimator must be reasonably efficient.

We also note that it is difficult to compare the statistical efficiencies of estimates based on bicast or other flexicast experiments with those based on an omnicast experiment as they are not on equal footing. If the total number of probes is fixed, the total amount of expected traffic is different for omnicast and flexicast experiments. Even if we fix the total amount of expected traffic on all links under the schemes, the different links will have different expected probe sizes. It is not possible to make the expected number of probes in each link be the same under the different schemes. Moreover, omnicast experiments contain information about all higher-order moments while the flexicast experiments are designed to sacrifice the higher-order moments to reduce data complexity.

To keep the comparisons meaningful, we will examine the efficiencies of estimation methods based on omnicast and bicast experiments separately. The comparisons here are based on a three-layer symmetric binary tree. The link delay distributions were chosen to be truncated geometric with five bins.

Figures 3.5, 3.6, and 3.7 shows the performance of the full EM-based MLE, PLE, and the polynomial estimator for the last bin for three links: $\alpha_1(4), \alpha_2(4)$, and $\alpha_4(4)$. The size of the omnicast experiment was 20,000 total probes. Recall that α_1 is the first link on the tree while α_4 corresponds to one of the receiver nodes. The figure suggests that the bias is small (medians close to the true values). The performance of the PLE is close to that of the MLE on links 1 and 4 with IQR ratios of 1.01 and 1.06. The performance on the interior link is somewhat worse with an IQR ratio of 1.38. This conclusion seems to be true in general; i.e., the relative performance of the PLE gets worse as one moves to the interior of the tree. So we would expect the

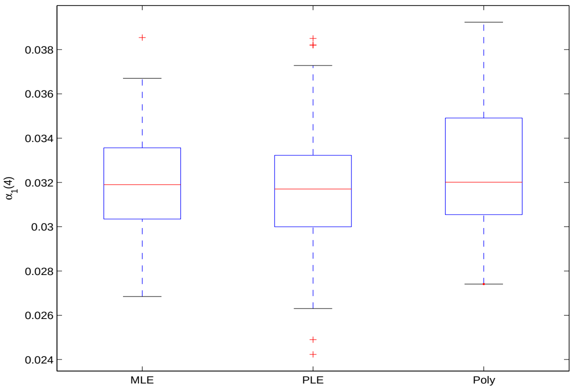

Figure 3.5: Boxplot of the estimates for $\alpha_1(4)$ for the multicast-based estimators.

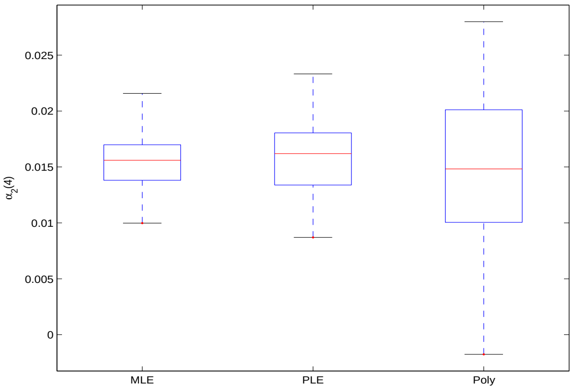

Figure 3.6: Boxplot of the estimates for $\alpha_2(4)$ for the multicast-based estimators.

performance to be poor for internal nodes in the middle of a large tree with many layers. The polynomial estimator, on the other hand, is considerably less efficient in all three cases with IQR ratios of 1.36, 3.18, and 2.47. We also compared the performance of the estimators for other bins and links. In general, the performance of the polynomial estimator is quite good for the first link ($\vec{\alpha}_1$) but gets progressively worse as we move deeper down the tree. This is because the polynomial estimator uses a lot of the data in estimating the first link but uses less and less data as we move down.

For the bicast-based estimators, we considered two different experiments: (i) all possible pairs and (ii) *min+1*. The estimators include all-pairs MLE, all-pairs grafting, *min+1* MLE, and *min+1* grafting. The comparisons of the estimators for the

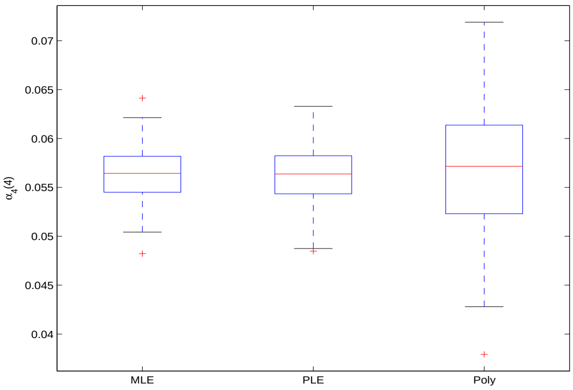

Figure 3.7: Boxplot of the estimates for $\alpha_4(4)$ for the multicast-based estimators.

first and last bins are shown in Figures 3.8, 3.9, and 3.10 and 3.11, 3.12, and 3.13 respectively. In terms of precision, the *min+1* MLE is very comparable to the all-pairs MLE except for estimating $\alpha_2(4)$. For $\alpha_4(4)$, it is actually slightly better. This can be explained by the fact that the *min+1* experiment allocates more probes for a scheme, $\langle 4, 5 \rangle$, that isolates link 4 thereby giving us a more precise estimate for this link. The grafting algorithms do slightly worse than the MLEs. For the all-pairs experiment, the IQR ratios of the grafting estimates to the MLEs for the zero bin on links 1, 2, and 4 are 1.267, 1.0231, and 1.4053. For the four bin, they are 1.2309, 2.1632, and 1.5634. For the *min+1* experiment, the IQR ratios for the zero bins on links 1, 2, and 4 are 1.4448, 1.6537, and 1.1899. For the four bin they are 1.7956, 1.4418, and 1.7397. In general the less precise algorithm do not perform as well in the interior of the tree at the tails of the distributions. Again, there is not a significant advantage of all-pairs experiment over the *min+1*.

3.4 Optimal Allocation of Probes

We now turn to an important issue in designing flexicast experiments, viz., how to optimally allocate the number of probes among the various schemes within a flexicast experiment. We do a limited study based on binary, symmetric trees and bicast

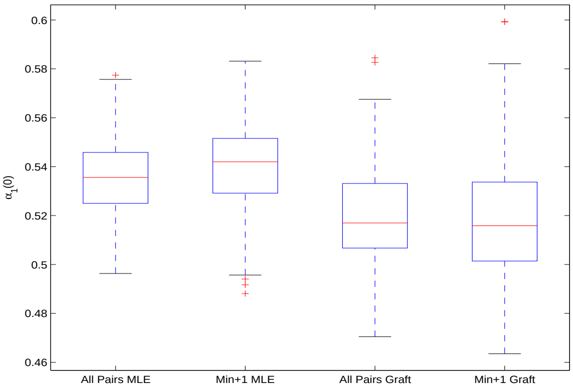

Figure 3.8: Boxplot of the estimates for $\alpha_1(0)$ for the bicast-based estimators.

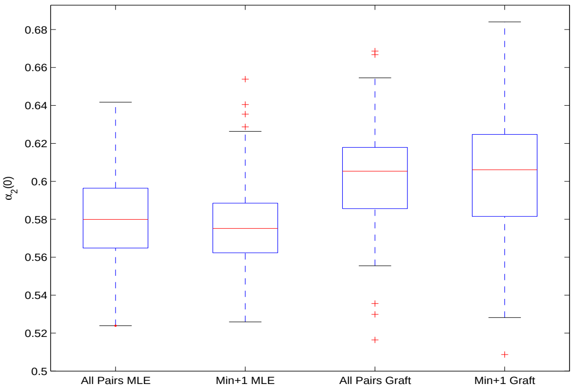

Figure 3.9: Boxplot of the estimates for $\alpha_2(0)$ for the bicast-based estimators.

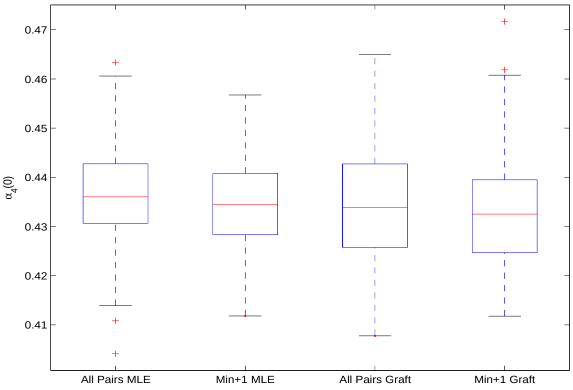

Figure 3.10: Boxplot of the estimates for $\alpha_4(0)$ for the bicast-based estimators.

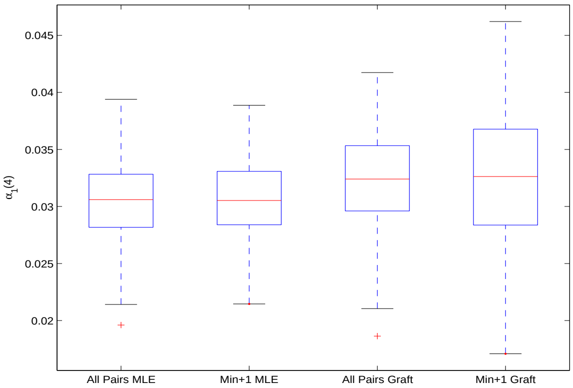

Figure 3.11: Boxplot of the estimates for $\alpha_1(4)$ for the bicast-based estimators.

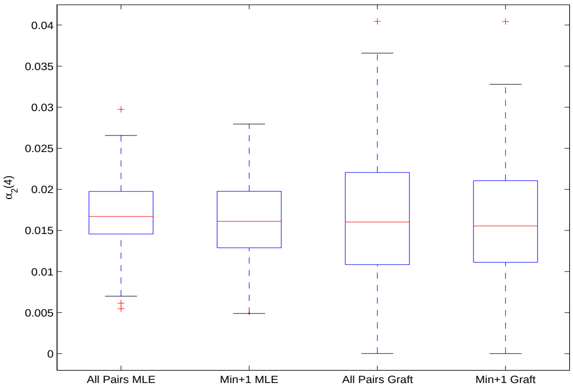

Figure 3.12: Boxplot of the estimates for $\alpha_2(4)$ for the bicast-based estimators.

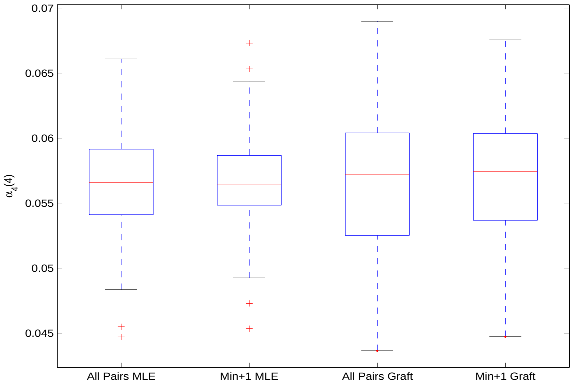

Figure 3.13: Boxplot of the estimates for $\alpha_4(4)$ for the bicast-based estimators.

schemes. A comprehensive study will involve a lot of additional work as the optimal allocations depend on the values of the unknown delay distributions and the tree topology. This is called *local optimality* in the design literature. The investigation here is intended to develop some preliminary insights and also suggest how one could go about studying the problem in general.

The question of interest is the following: given a fixed budget of probes, how should they be allocated among the various flexicast schemes?

We conduct our study on a three-layer binary, symmetric tree. For the link delay distributions, we use geometric distributions truncated to five bins. Based on our experience with real data and the network simulator, the geometric distribution is a reasonable choice with its large mass at zero and decaying tails. We let the parameter of the distributions range from $p = 0.1$ to $p = 0.9$ with a step size of 0.1. This range allows us to consider very good links with light tails ($p = 0.9$) and more congested links with heavier tails ($p = 0.1$).

We will use the *min+1* bicast experiment: $\mathcal{C} = \{\langle 4,5\rangle, \langle 6,7\rangle, \langle 5,6\rangle, \langle 4,7\rangle\}$. Note that the last two probes split at node 1 while first two probes split at a lower level. We view links 1, 4, 5, 6, and 7 as edge links while links 2 and 3 will be considered internal/back bone links. Links of the same type (edge, backbone) will have the same distribution. Based on this symmetry, the optimal proportion of probes sent $\langle 4,5\rangle$ and $\langle 6,7\rangle$ should be equal and that to $\langle 5,6\rangle$ and $\langle 4,7\rangle$ should be equal. Let τ refer to the total proportion of probes sent to first group; then the second group will get a proportion $1 - \tau$. The design problem is to identify the optimal value of τ.

The criterion we use here is D-optimality, commonly used in the experimental design literature. Specifically, the optimal value of τ is obtained by maximizing the determinant of the Fisher information matrix. As noted before, this value depends on the unknown parameters of the delay distributions, in addition to the tree topology. This is referred to as local D-optimality. In this problem, one cannot obtain universal results for all distributions. Nevertheless, it is useful to examine how τ varies in selected situations in order to gain some insights. To implement the optimal allocation in practice, one would have to use prior knowledge of the distributions.

The optimal values of τ are shown in Figure 3.14. The left panel corresponds to the case where all the link-delay distributions are the same. The x-axis corresponds to the failure probability p in the geometric distribution. (Note that a high value of p implies higher probability of smaller delay). Interestingly, the optimal value of τ is constant (around .75) as p ranges from .1 to about .8 and then decreases slightly to about 0.7 as p increases to 0.9. Based on this, the bicast pairs $\langle 4, 5\rangle$ and $\langle 6, 7\rangle$, which split at a lower level in the tree, should get about $35 - 37\%$ of the probes each while the bicasts $\langle 5, 6\rangle$ and $\langle 4, 7\rangle$, which split at a higher level, should receive only about $12 - 15\%$ of the probes. Note the pairs that split at the lower level provide the most information for estimating the receiver links. Further, the total number of probes at each link varies: under the above optimal setting, all the probes pass through the link at the top layer, links in layer two (the "backbone" links) each see about 3/4 of the probes, and those at layer three (receiver links) each see only 1/4 of the probes.

The right panel in Figure 3.14 shows the optimal allocations for a two-dimensional situation: the edge links (link 1 and the receiver links) have the same truncated geometric distribution with "failure" probability p_1 (x-axis) while the backbone links 2 and 3 have the same distribution with probability p_2 (y-axis). The z-axis shows the values of τ, the optimal allocation. The left panel corresponds to the diagonal line on the x-y axis. For most of the $p_1 - p_2$ values, the optimal value of τ is between $0.6 - 0.8$, again indicating that a higher proportion should be sent to the bicast pairs that split at the lower level. The exception is when the failure probability p_2 of the backbone links become larger than about 0.8 and p_1 is in the range $0.1 - 0.5$, the values of τ decrease, implying that the pairs that split at a higher level should get a larger proportion of the probes. This is likely due to the higher congestion on link 1 as compared with the backbone links. Probes splitting at node 1 help to provide a good estimate of link 1 which is important in this case for sorting out the links below it.

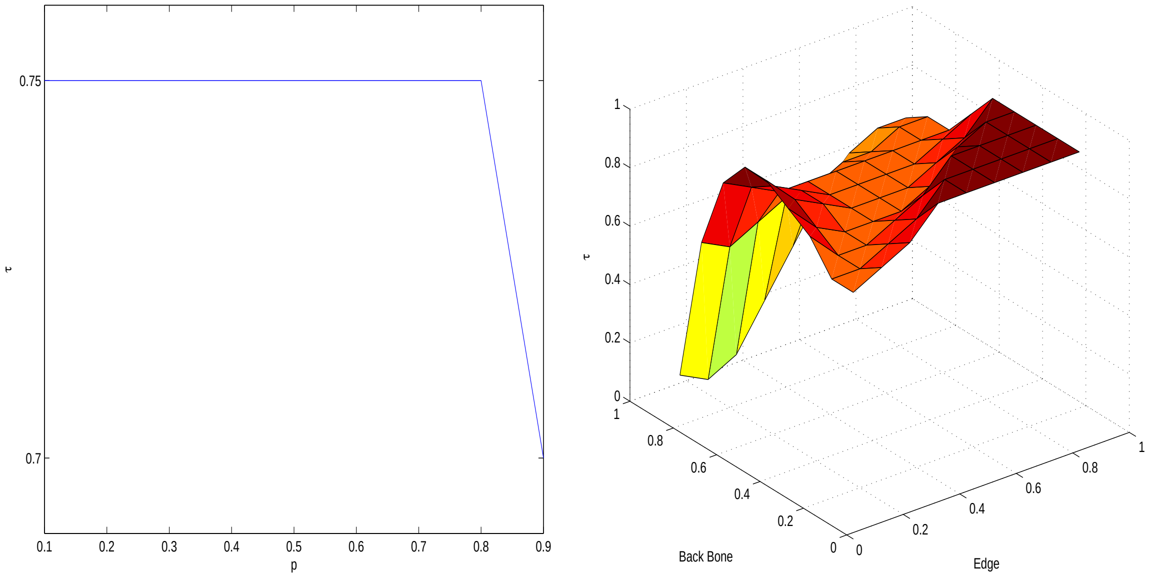

Figure 3.14: The optimal allocation of probes will all links the same (left panel) and the optimal allocation of probes with interior links the same and edge links the same (right panel).

3.5 Simulation Studies

In this section, we use simulation to assess the performance of the estimation methods under two scenarios. The first is done under the stochastic model assumptions in Section 2.3. The second uses a more realistic framework using the ns-2 network simulator.

3.5.1 Model-Based Simulation

Our simulation studies showed that if the true link-level distributions are discrete, the MLE as well as grafting methods are able to recover the link-level estimates well without any bias. When the true distributions are continuous, however, the binning seems to introduce some bias. The problem arises from the fact that the end-to-end data are grouped into bins, so we have discretized sums instead of sums of discrete values from each link. The extent of the bias depends on the bin size, the link, and other variables.

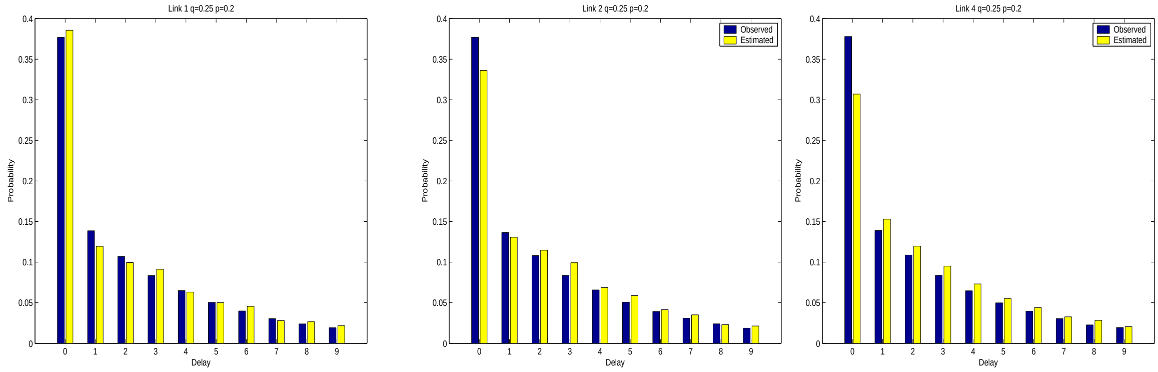

Figure 3.15: Observed and estimated distributions for links 1, 2, and 4 showing bias when the estimation is applied to binned end-to-end data. The distributions are exponential with mean 1 mixed with point mass at 0 with probability .2. The bin size is .25

To develop some insights, we considered a three-layer symmetric binary tree, Figure 3.4, and focussed on the MLE for a minimum bicast experiment. Each link distribution was taken to be a mixture of exponential with mean one and point mass at 0. The point mass, corresponding to no delay, is common in many real situations. Various bin sizes and point mass probabilities were considered in the study. Figures 3.15 and 3.16 give selected results. Figure 3.15 show the results from links 1, 2, and 4 (link 3 has same behavior as 2 and links 5,6,7 have same behavior as 4) with a bin size of $q = 0.25$ and a point mass with probability $p = 0.2$. Figure 3.16 shows the

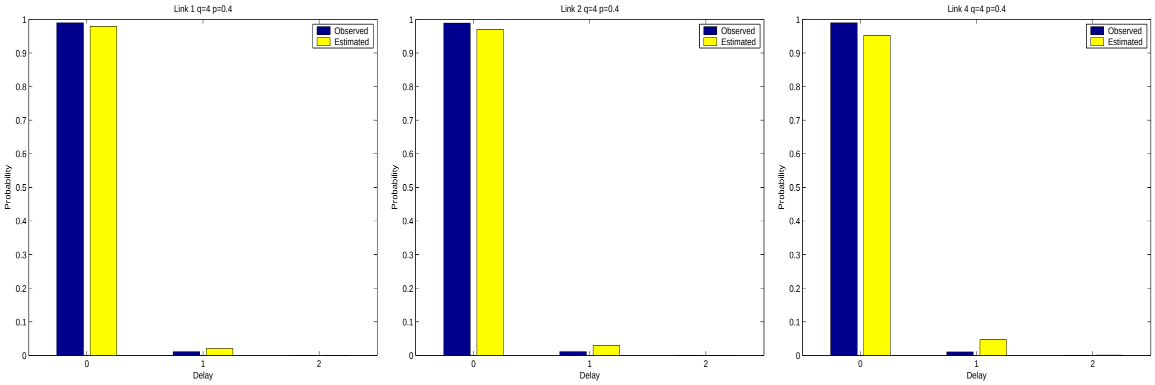

Figure 3.16: Observed and estimated distributions for links 1, 2, and 4 showing bias when the estimation is applied to binned end-to-end data. The distributions are exponential with mean 1 mixed with point mass at 0 with probability .4. The bin size is 4

results for the same links with a bin size of $q = 4$ and a point mass with probability 0.4.

The largest bias occurs for the zero bin ($\alpha_k(0)$); larger point-mass probabilities or bin sizes lead to smaller bias. For instance, for a bin size of $q = 0.25$ and point-mass probabilities of 0.1, 0.2, and 0.4, the underestimate of the zero probability on link 4 is 34%, 24%, and 9% respectively. For bin size of $q = 4$, the corresponding underestimates reduce to 8%, 6%, and 4%. Note also from Figures 3.15 and 3.16 that the bias is greatest for the receiver node (link 4) and decreases as one goes up to the source node. For example, with bin size $q = 4$ and point mass-probability fixed at 0.4, the zero bin on links 1, 2, and 4 are underestimated 1%, 2%, and 4% respectively.

The effect of this bias on the other bins is much smaller. For the most part, the bias at the zero bin seems to be spread across the rest of the distribution. The estimate at bin 1 seems to compensate somewhat more than the other bins. There is also some compensation across links; the zero bin for link 1 is overestimated while the zero bins for other links are underestimated.

We are currently exploring some methods for bias correction. One possible approach is the use of parametric bootstrapping as follows: Fit a continuous distribution to the estimated histograms of the links (with bias), simulate data from these distributions to get both observed and estimated values (as in Figures 3.15 and 3.16) and use the estimated bias to do bias correction. There are some problems with this approach (the estimated histograms and hence the continuous distributions are biased) and have to be studied further.

3.5.2 Network Simulation

We now examine the performance of the proposed estimators in a realistic network environment by using the ns-2 (Information Sciences Institute (2004)) simulation package. This allows one to construct any topology and generate traffic and transmit packets using real network protocol. It gives users control over the hardware and software aspects of a network including bandwidth, propagation delay, traffic volume, and traffic protocol. We constructed the topology shown in Figure 2.3 to mimic the UNC network. For links between core routers, we used 500 megabit links and for links to endpoints, we used 50 megabit links. Background traffic on the core links consists of 27 TCP connections and 5 UDP connections. TCP connection acknowledge reception of packets by the receiver. Lost packets are retransmitted by the sender and result in a slower transmission rate. Therefore, TCP connections are responsive to congestion. UDP connections do not have any of the above features and continue to send packets at a constant rate, thus being unresponsive to congestion patterns. On the edge links, the background consists of 6 TCP connections and 1 UDP connection. The probe traffic consists of 40 bit UDP packets using the multicast protocol.

Every one tenth of a second, the probing mechanism selects a scheme at random from $\mathcal{C} = \{\langle 4,5\rangle, \langle 6,7\rangle, \langle 8,10\rangle, \langle 11,12\rangle, \langle 13,14\rangle, \langle 15,16\rangle, \langle 17,18\rangle\}$ and sends a packet to its receivers. Probing lasts for 700 seconds resulting in about 1000 packets sent to each pair. This is approximately the length of a session that we would use for monitoring a real network. The end-to-end delays are discretized using a bin size of $q = .00005s$. This is an extremely fine scale resulting in a maximum link delay setting of $b = 155$. Furthermore, the ns-2 package allows us to record the true link delays and hence directly obtain the link delay distributions for verification.

Figures 3.17 and 3.18 show the fitted distributions along with the observed distributions for selected links. We see the effect of the bias from binning discussed in the last section. Aside from this, the estimation procedure does a very good job of capturing the distribution despite a violation of the temporal and spatial independence assumptions in the model. The results show that the estimation procedure performs well in a real network setting.

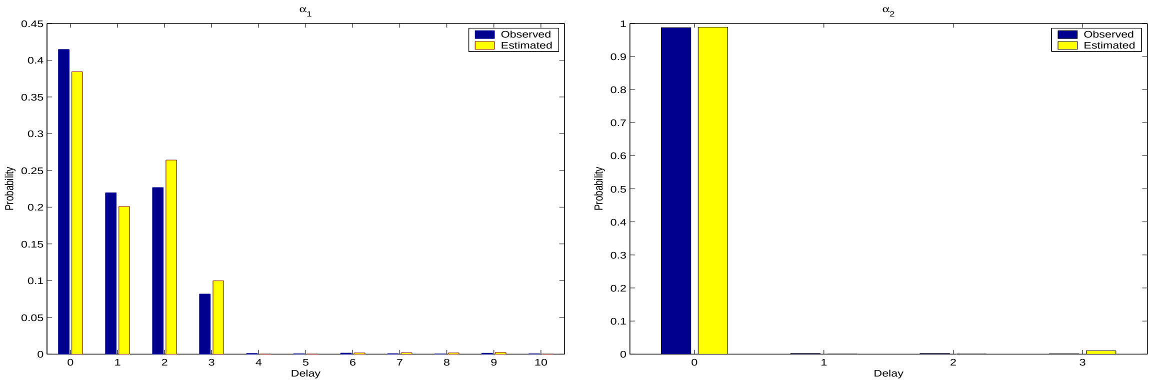

Figure 3.17: Observed and estimated delays for links 1 and 2 of the ns-2 example.

3.6 Application to the UNC Campus Network Data

One of the main goals of this project is to evaluate the UNC campus network for VoIP capabilities. As discussed before, this real-time application requires excellent link quality in order to be successful. In particular, the presence of any large delays

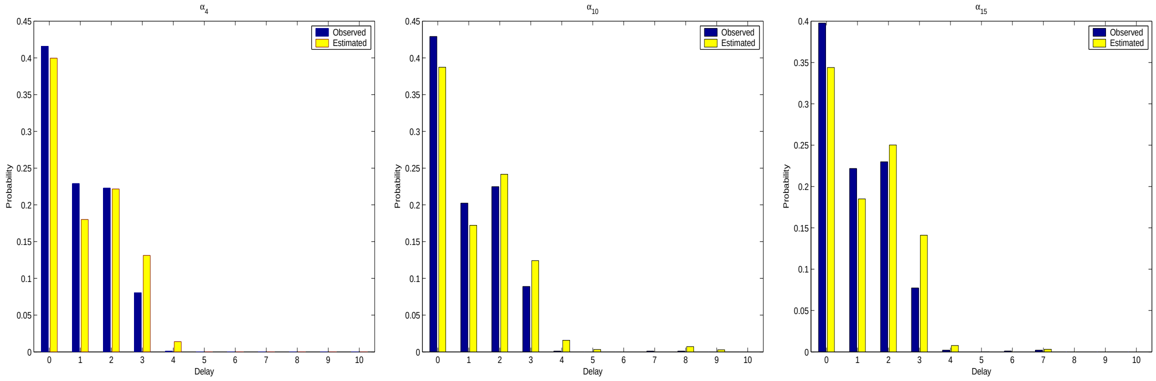

Figure 3.18: Observed and estimated delays for links 6, 11 and 15 of the ns-2 example.

can significantly reduce the quality of the phone calls. In this section, we validate the usefulness of the methods developed in the paper by applying them to real data. We have collected extensive amounts of data but report here only selected results based on data collected at two-hour intervals starting at 8:00 a.m. and ending at 2:00 a.m. on a typical school day.

The data were collected using a tool designed by Avaya Labs for testing a network's readiness for VoIP. There are two parts to the tool. First, there are the monitoring devices that are computers deployed throughout the network with the capability of exchanging VoIP-style traffic. These devices run an operating system that allows them to accurately measure the time at which packets are sent and received. The machines collect these time stamps and report them back to the second part of the system: the collection software. This software remotely controls the devices and determines all the features of each call, such as source-destination devices, start time, duration, and protocol which includes the inter-packet time intervals. The software collects the time stamps when the calls are finished and processes them. The processing consists of adjusting the time stamps to account for the difference among the machines' clocks, and then calculating the one-way end-to-end delays.

For the data collection, Sitterson served as the root, and we used seven bicast pairs to cover the 14 receiver nodes:

$$\mathcal{C} = \{\langle 4,5\rangle, \langle 6,7\rangle, \langle 8,10\rangle, \langle 11,12\rangle, \langle 13,14\rangle, \langle 15,16\rangle, \langle 17,18\rangle\}. \tag{3.20}$$

The network only allowed unicast transmission protocol, so back-to-back probing was used to simulate multicast transmissions. The span of time between the two packets comprising the back-to-back probe was on the order of a few nanoseconds while the time between successive probes was one tenth of a second. Prior experimentation using the call synthesis tool and this probing method leads us to believe that the correlation between the two packets on the shared links is close to one. Each probing session consisted of two passes through the pairs in the order presented. On each pass, each pair was probed for 50 seconds. Thus, we have 1000 probes for each pair during each monitoring session. Maximum likelihood estimation was used to deconvolve the link distributions.

In this section, we present results related only to discovering links that have significant probabilities of large delays. For this reason, we used a bin size of $q = .0002s$. Above this threshold, delays can become detrimental to call quality. We expect that most links in this network would have distributions with most of the mass on the zero bin. Nonetheless, a mass as low as .01 on the rest of the delays could prove troublesome.

Figure 3.19 shows the probability of delays larger than .0002 seconds at various times throughout the day. First, note that the links between the main core routers are of very high quality. The Sitterson outgoing link is also extremely good. The rest of the links do experience some congestion, varying over the course of the day. Many of the school buildings show a diurnal effect with lots of activity contributing to higher delays starting around noon and continuing throughout the afternoon. In particular, there is a bit of a spike around 4:00 p.m. This spike is evident on link 9 which is a larger core-to-core router. The dormitory links show more consistent traffic throughout the entire day with some elevated delays in the later evening. Links that show 1% or more large delays would likely require an upgrade in order to be able to handle the increased load placed upon them by VoIP which uses a more aggressive protocol than the prevalent TCP traffic. Several receiver links already show close to 5% large delays without a strong VoIP presence. Even the large link 9 seems to be problematic as it needs to perform almost flawlessly because it handles considerably more traffic than the receiver links.

To look at some aspects of the analysis in more detail, we solved the inverse problem using a bin size of $q = .00002s$ for the time periods 12:00 p.m. and 10:00 p.m. This gave us ten times the resolution of the above analysis. Further, it allowed us to break down the previous analysis to see where the delays fall within the smallest bin. Figures 3.20, 3.21, and 3.22 show the first 20 bins of these detailed results for Davis Library, South Building, and Hinton Dorm respectively. The first thing to note is that most of the mass is still on the lowest bins so the vast majority of packets experience very little delay. Both Davis and South exhibit a strong diurnal effect. Unlike the dorm, the traffic in these buildings dies off late at night.

The analyses of other similar data sets collected on the network over a period of time showed remarkable stability in the results and conclusions. These delay probabilities indicated to the UNC IT group that the current network is not capable of supporting the VoIP application. From our point of view, the results are qualitatively consistent with the overall behavior that is to be expected for this network. This serves as a validation of the techniques of the techniques studied here, both from statistical and implementation perspectives.

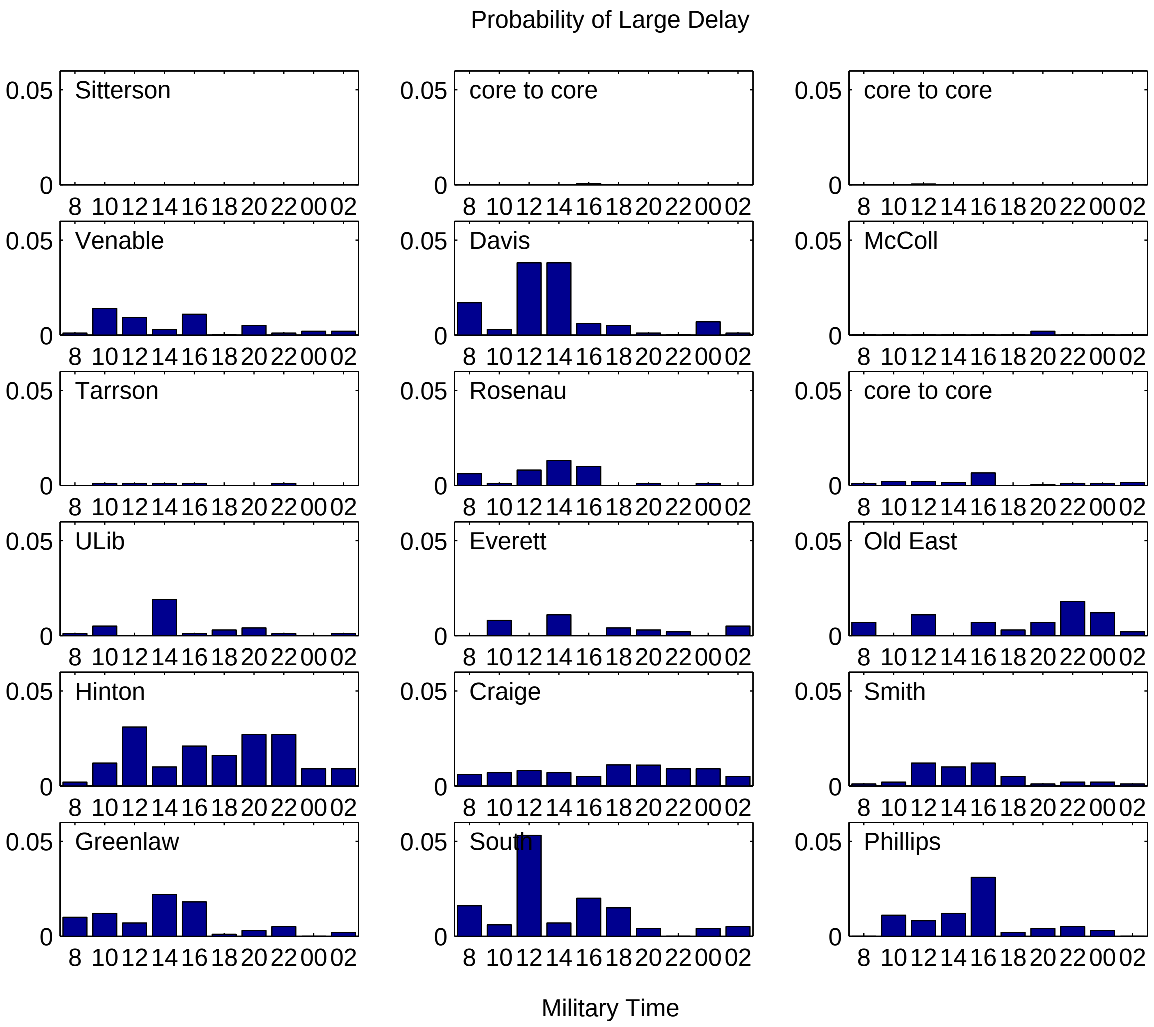

Figure 3.19: Probability of large delay on each link throughout the day.

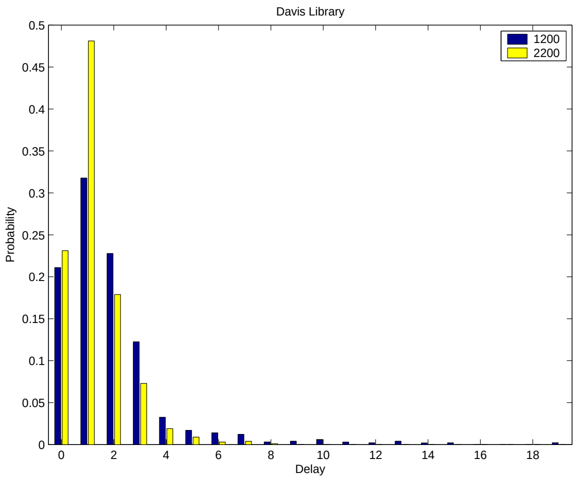

Figure 3.20: Delay distribution for Davis Library at 12:00 p.m. and 10:00 p.m.

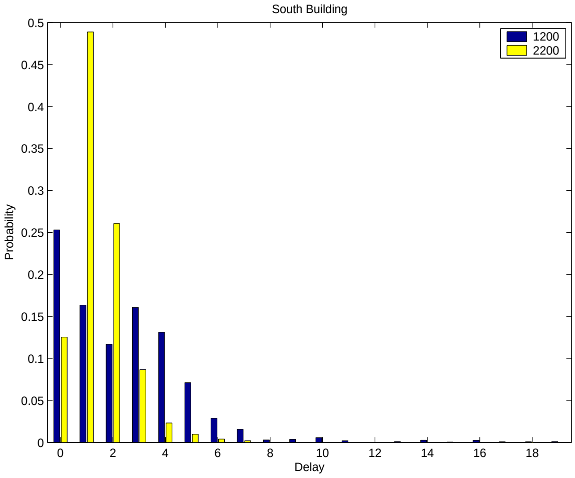

Figure 3.21: Delay distribution for South Building at 12:00 p.m. and 10:00 p.m.

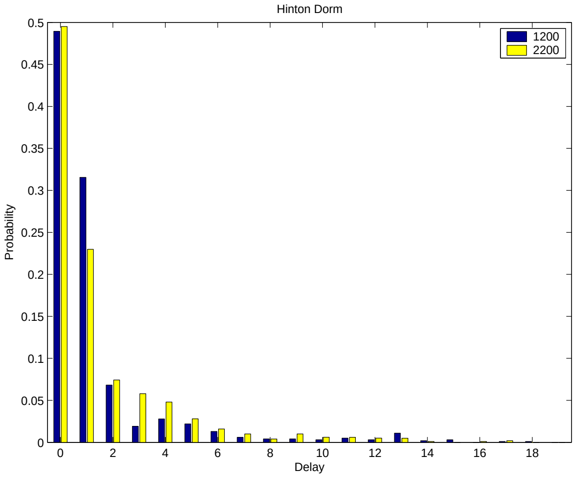

Figure 3.22: Delay distribution for Hinton Dorm at 12:00 p.m. and 10:00 p.m.

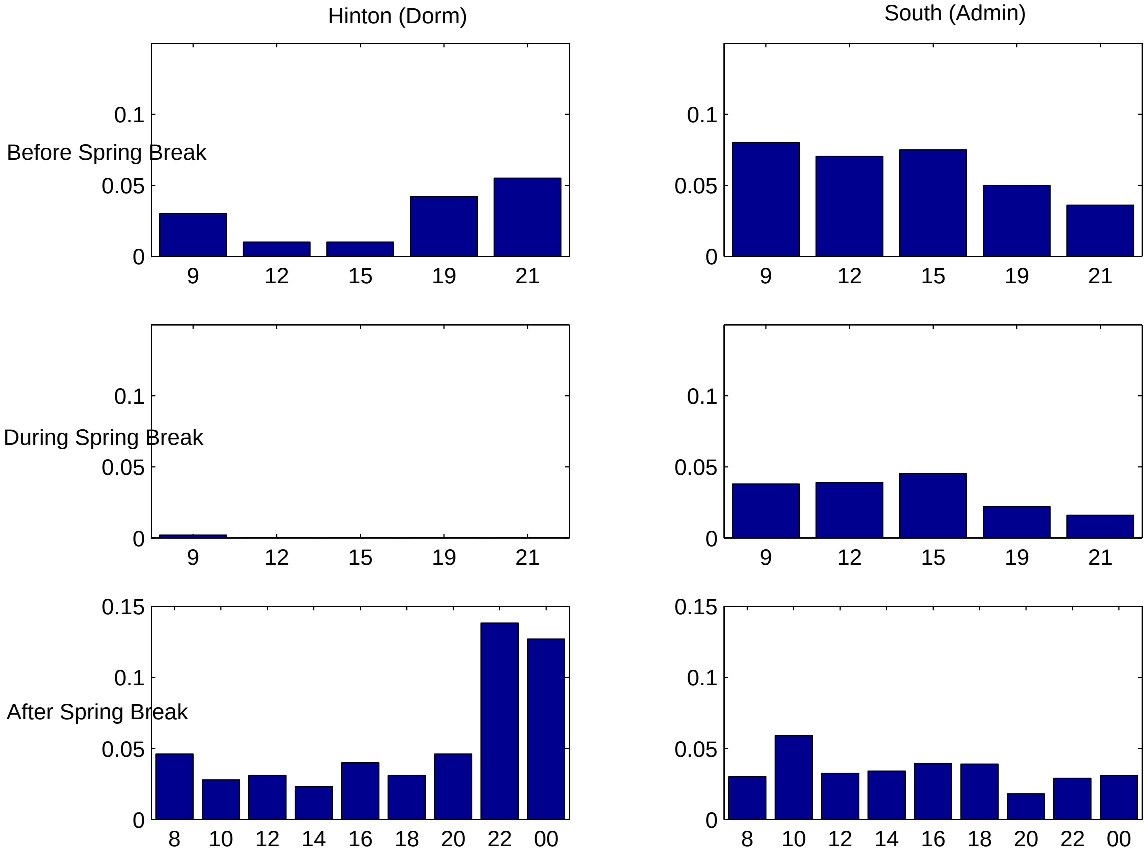

Figure 3.23: Before, during, and after probabilities for large delays at a dorm link and university office link.

For additional verification, we also collected data before, during, and after the 2005 spring break at UNC. These results are shown in Figure 3.23. Notice that there were no large delays at the dorm link during the break since all the students are away from campus. The office link continued to display the usual delay patterns.

3.7 Bayesian Inference

3.7.1 Estimation

One can also use Bayesian methods with data augmentation for inference. Data augmentation is similar to the EM algorithm in that the we obtain the missing data and use it to obtain the parameters. Under a data augmentation scheme, we draw the missing data according to its conditional distribution given some value of the parameters and the observed data. Then, given these values of the missing data and a prior distribution on our parameters, we draw the parameters from their posterior distribution. When the sequence converges, we can use our sample of parameter draws to estimate various quantities of the posterior distribution such as the mean and variance.

The details are as follows. We begin by setting a prior for the parameters $\vec{\alpha}_k$. The conjugate prior for the multinomial is Dirichlet:

$$p_k(\vec{\alpha}_k) \propto \prod_{i=0}^{b} \alpha_k(i)^{u_i - 1}; \ \sum_{i=0}^{b} \alpha_k(i) = 0, \ u_i > 0 \ \forall i. \tag{3.21}$$

This prior is quite flexible as it can be used in both diffuse and informative situations. The expected value of any particular $\alpha_k(i)$ is given by $u_i/(\sum_{j=0}^{b} u_j)$ so the u_i can be chosen to center on any link distribution that we desire. Additionally, the variance of this distribution depends largely on the $\sum_{j=0}^{b} u_j$ so we can make the distribution as focused or diffuse as we desire simply be specifying larger or smaller u_i which can be thought of as data counts from a previous experiment. Finally, the prior can be made completely flat by setting all the $u_i = 1$ which will give us a noninformative prior. The prior is also quite convenient as it is the conjugate prior for the observed likelihood.

The data augmentation step is similar to the E-step in the EM algorithm. For a given end-to-end outcome, we want to take a random draw of the individual link delays that give rise to that end-to-end outcome. For each observation y_i, we begin be determining all of the link delay combinations that lead to y_i. Using the current value of $\vec{\alpha}$ from the previous draw, we compute the probabilities of each of these link delay outcomes. Dividing each probability by their sum, we obtain the required conditional probability to perform the augmentation. With the missing data, we can construct counts $M_{k,i}$, the number of times that link k had a delay of i. This leads to a very simple draw from the posterior of alpha which is again Dirichlet:

$$f_k(\vec{\alpha}_k) \propto \prod_{i=0}^{b} \alpha_k(i)^{M_{k,i}+u_i-1}. \tag{3.22}$$

This algorithm has a number of attractive qualities. Given the missing data, the $\vec{\alpha}_k$ are independent of each other. As a result, convergence to the stationary distribution is extremely fast. Additionally, since the parameter space is fairly small and each vector must sum to one, any valid starting point is relatively close to the high probability region of the true posterior. Finally, the highly structured relationship between an end-to-end observation its valid link delay combination seems to insure that extreme draws are avoided in the augmentation step. As a result of all this, valid inferences about the posterior distribution can be made with a small number of draws. A practical discussion of convergence and inference issues is given below along with some numerical investigation.

One advantage of using this Bayesian estimation procedure its ease of inference. To construct confidence intervals and perform tests, we need estimates of the covariance matrix of the parameters. In the maximum likelihood setting, this requires us to construct and invert the observed Fisher information matrix. This can be quite tedious if the tree or number of bins is large because the Fisher is then large and difficult to construct and manipulate. The Bayesian sampler described above avoids this difficulty because we can use an empirical estimate of the mean and covariance of the posterior distribution to do inference. Because of the ergodicity of the Markov chain, the mean of the posterior can be easily computed by taking the mean of the chain after removing the burn-in portion. The MCMC literature is filled with methods for computing estimates of the covariance matrix. One simple method is to keep every k-th sample from the Markov chain and then treat this as an *i.i.d.* sample. The value of k should be chosen so as to reduce the autocorrelation to some ignorable amount. This process can be vastly simpler than the Fisher information manipulation required for MLE inference. If the prior is chosen to be noninformative, then there should be no resultant philosophical issues.

3.7.2 Numerical Example

We performed model based simulation with data generated according to our assumptions in order to test the performance of the Bayesian estimator. We generated data from a three-layer tree with truncated geometric distributions on the links. The edge links were given ten-bin truncated geometric distributions with $p = .5$ and the interior links with $p = .8$. These distributions were chosen in order to be similar to those displayed in our ns-2 simulations. We used the noninformative prior described above. Figure 3.24 shows the parameter draws for 1,000 iterations of the DA scheme for the first three bins on links 1 and 2. These traces are indicative of the edge and interior links respectively. The straight lines on the plots are the parameter values from which the data were generated.

The plots indicate the the Markov chain seems to converge fairly quickly, in about 50 iterations. Our experience with other distributions indicates that this is pretty typical. As a result, good inference can be made about the posterior with significantly fewer draws than are displayed here. Disregarding the first 50 iterations as burn-in, the posterior mean estimated from the first 200 iterations is very close to the mean from the entire trace. Furthermore, the posterior mean is quite close to the known true value.

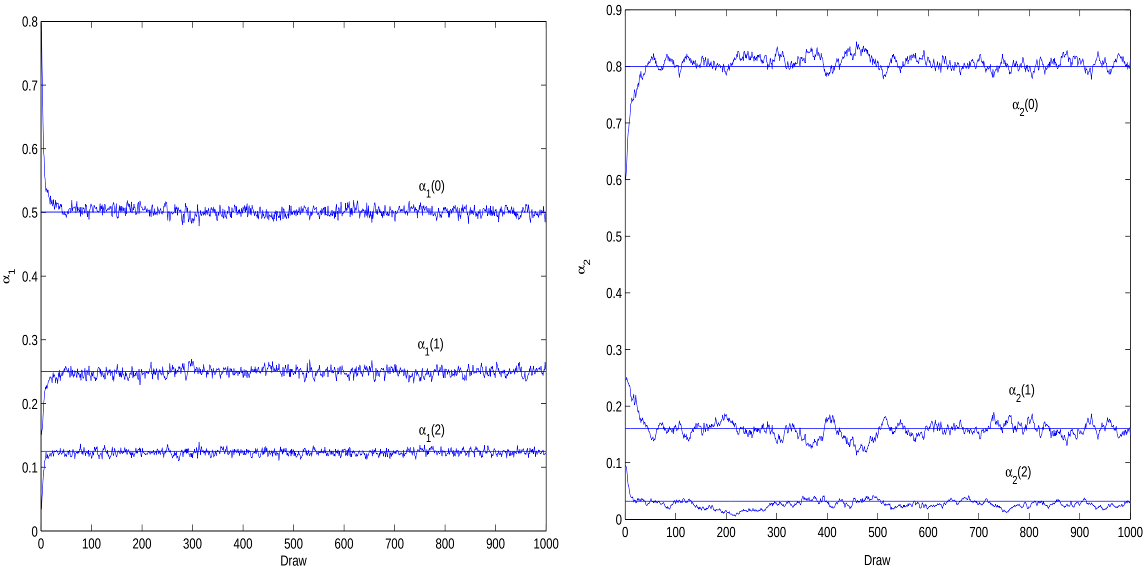

Figure 3.24: Data augmentation parameter draws for the first three bins on links 1 and 2.

CHAPTER IV

Continuous Delay Modeling

This chapter considers the modeling the distributions as continuous, possibly with point masses at zero and infinity.

4.1 Identifiability

In the last chapter, we examined identifiability conditions for the discrete model and showed that flexicast probing in which every internal node is a splitting node is necessary and sufficient for the recovery of all link delay distributions. In this section, we expand the discussion to consider continuous distributions.

In general, pure unicast probing is insufficient for the estimation of link characteristics based on end-to-end measurements. To see this, consider the estimation of the link-level cumulant generating functions (CGFs) based on the end-to-end cumulant generating functions. These functions are convenient since the end-to-end functions are a linear combination of the link functions:

$$\Psi_Y(t) = A\psi_X(t), \tag{4.1}$$

where Ψ_Y are the CGFs of the end-to-end functions, ψ_X are the link CGFs and A is the $|R| \times |E|$ routing matrix where $A(i,j) = 1$ if the path $\mathcal{P}_{0,i}$ includes link j. The columns of the matrix are always linearly dependent since the first column is a column of ones and the sum of all the receiver columns is also a column of ones. The problem can be solved only by assuming that the link delay distributions are all identical. Although not a formal proof, this demonstrates the main difficulty of pure unicast probing: the end-to-end measurements will always be a non-invertible linear combination of the link measurements.

In general, flexicast probing is required because it gives joint information about the end-to-end measurements that relates to the shared path. It is easy to see that each internal node must be used as a splitting node at least once. More probing schemes can be added, but for many problems the additional schemes act more like repeated measurements rather than measurements taken at new locations.

We start with a simple sufficiency condition that covers many continuous parametric distributions that are appropriate for delay modeling. The key feature here is that central moments, $\mathrm{E}(X_k - \mu_k)^h$, are estimable for $h > 1$ on all links in the tree. Thus we can estimate any distribution that is identifiable from these moments.

Proposition IV.1. *Let $\mathcal{T}$ be a general tree network. Let $\mathcal{C}$ be a collection of k-cast schemes $\mathcal{C}_j,\ j = 1, ..., C$. Let each link delay distribution be given by $F_k(x; \vec{\theta}_k)$. The parameters of the link-level delay distributions are identifiable if (a) For every internal node $s \in \mathcal{I}$, there is at least one k-cast scheme $\mathcal{C}_j \in \mathcal{C}$, with $k > 1$, such that s is a branching node for $\mathcal{C}_j$, (b) every receiver $r \in \mathcal{R}$ is covered by at least one $\mathcal{C}_j \in \mathcal{C}$, and (c) for $k \in \mathcal{I}$, $\vec{\theta}_k$ is estimable based on central moments of order two and higher and for $k \in \mathcal{R}$, $\vec{\theta}_k$ is estimable from the ordinary or central moments of any order.*

Remark 1: This proposition gives identifiability for many useful parametric distributions including exponential, gamma, log-normal, Weibull, and others. The key is that each of these distributions has higher order moments that are functions of or provide information about the mean.

Remark 2: This proposition allows the estimation of different distributions on different links. This provides a great deal of modeling flexibility. Large links can be given one distributional form and smaller links another. Further specification can be considered according to prior knowledge as long as estimability is still achieved.

Proof: The proposition is easily shown by demonstrating that the required moments are available under this probing scheme.

Consider link 1. By assumption, there is some scheme $\mathcal{C}_j$ with receivers k and l that split at node 1. Under our assumptions, the covariance of the end-to-end delays observed at these nodes is given by the variance of the delay distribution on link 1. Note that we also get the variances of the path-level distributions on $\mathcal{P}_{1,k}$ and $\mathcal{P}_{1,l}$.

We proceed by induction. Assume that we have estimated the central moments of orders two through $h-1$ for link 1 and the paths to the receivers, $\mathcal{P}_{1,k}$ and $\mathcal{P}_{1,l}$. Consider the quantity:

$$\mathrm{E}(Z_k^{\lfloor h/2 \rfloor} Z_l^{h-\lfloor h/2 \rfloor}), \tag{4.2}$$

where Z is a centered end-to-end measurement. This estimable quantity is a function of the h-th order central moment of link 1 and the lower order moments of link 1 and the receiver paths. Thus central moments of all order two and higher are estimable. Thus, we can estimate all the components of $\vec{\theta}_1$.

The rest of the links in $\mathcal{I}$ are handled through induction with peeling. For any node s, there is once again some pair k and l that splits at s. If we have estimated the required moments for all ancestors of s, then the moments for s are obtained by estimating the central moments of the distribution on $\mathcal{P}_{0,s}$ (in the same fashion that we used for link 1) and peeling off the moments of each ancestor link.

Receiver links are also similar except that we can use the first order moment as well. Clearly we can estimate the mean of the end-to-end path. Since we have completely characterized the distributions of all the links in $\mathcal{I}$, we can estimate their means and peel them from the end-to-end mean. □

To clarify, consider an example. Let $\mathcal{T}$ be a three-layer binary, symmetric tree with probing experiment $\mathcal{C} = \{\langle 4,5\rangle, \langle 5,6\rangle, \langle 6,7\rangle\}$. Let the delay on link k be distributed $Gamma(\alpha_k, \beta_k)$. Based on covariance, we can get the following set of equations:

$$Cov(Y_5^{\langle 5,6\rangle}, Y_6^{\langle 5,6\rangle}) = \alpha_1\beta_1^2, \tag{4.3}$$

$$Cov(Y_4^{\langle 4,5\rangle}, Y_5^{\langle 4,5\rangle}) - Cov(Y_5^{\langle 5,6\rangle}, Y_6^{\langle 5,6\rangle}) = \alpha_2\beta_2^2, \tag{4.4}$$

$$Cov(Y_6^{\langle 6,7\rangle}, Y_7^{\langle 6,7\rangle}) - Cov(Y_5^{\langle 5,6\rangle}, Y_6^{\langle 5,6\rangle}) = \alpha_3\beta_3^2. \tag{4.5}$$

Let $E(Y_r) = \nu_r$. We also get the following equations based upon third moments:

$$E(Y_5^{\langle 5,6\rangle} - \nu_5)^2(Y_6^{\langle 5,6\rangle} - \nu_6) = 2\alpha_1\beta_1^3 \tag{4.6}$$

$$E(Y_4^{\langle 4,5\rangle} - \nu_4)^2(Y_5^{\langle 4,5\rangle} - \nu_5) - E(Y_5^{\langle 5,6\rangle} - \nu_5)^2(Y_6^{\langle 5,6\rangle} - \nu_6) = 2\alpha_2\beta_2^3 \tag{4.7}$$

$$E(Y_6^{\langle 6,7\rangle} - \nu_6)^2(Y_7^{\langle 6,7\rangle} - \nu_7) - E(Y_5^{\langle 5,6\rangle} - \nu_5)^2(Y_6^{\langle 5,6\rangle} - \nu_6) = 2\alpha_3\beta_3^3 \tag{4.8}$$

The terms on the left are all easily estimated based on their sample values. Estimators for $\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3,$ and β_3 are easily obtained by rearranging the above equations.

The parameters for the receiver links can also be estimated based on their second and third central moments, but they can also be estimated with just the first moments as well. Here are the equations for link 4:

$$E(Y_4) = \alpha_1\beta_1 + \alpha_2\beta_2 + \alpha_4\beta_4 \tag{4.9}$$

$$Var(Y_4) = \alpha_1\beta_1^2 + \alpha_2\beta_2^2 + \alpha_4\beta_4^2 \tag{4.10}$$

The unknown parameters are easily obtained from the observed values on the left and the estimated parameters on the right.

Often, we need to include a point mass at zero in order to capture the positive probability of a packet reaching an empty queue and experiencing no waiting time. The resulting distribution is a mixture of the point mass at zero and the continuous positive-valued distribution. Additionally, it is often useful to include a point mass at infinity and we will briefly discuss this in a later section.

The point mass turns out to be very useful for identifiability. A point mass allows us to isolate direct observations for each link that can be used for estimation.

Proposition IV.2. *Let $\mathcal{T}$ be a general tree network. Let $\mathcal{C}$ be a collection of k-cast schemes $\mathcal{C}_j,\ j = 1, ..., C$. Assume that X_k, the distribution on link k has the following distributional form:*

$$X_k = 0 \text{ w.p. } p_k > 0 \tag{4.11}$$

$$X_k \sim F_k(x) \text{ w.p. } 1 - p_k. \tag{4.12}$$

The link-level delay distributions are identifiable if (a) For every internal node $s \in \mathcal{I}$, there is at least one k-cast scheme $\mathcal{C}_j \in \mathcal{C}$, with $k > 1$, such that s is a branching node for $\mathcal{C}_j$ and (b) every receiver $r \in \mathcal{R}$ is covered by at least one $\mathcal{C}_j \in \mathcal{C}$.

Remark 1: Note that there are no conditions on the continuous part of the distribution F_k. This can be any parametric or nonparametric choice.

Remark 2: This proposition can easily be extended to consider point masses at different location or multiple point masses.

Proof: By assumption, there is a scheme with receivers k and l that split at node 1. Consider data from this scheme in which $y_k = y_l$. With probability one $x_1 = y_k = y_l$. Thus we have direct observation from link 1. In principle, we can now estimate the moment generating function (MGF) and compute the density of X_1.

We can similarly compute the path moments on $\mathcal{P}_{0,s}$ and use peeling to estimate the moments of X_s. Again we can estimate the MGF and thus the density. □

4.2 Maximum Likelihood Estimation

For parametric distributions, it is natural to consider maximum likelihood estimation. In order to discuss this setting, we begin with a simple example. Consider the simplest possible topology: the two-layer tree with two receivers. Let the delay distribution on each link have an exponential distribution with parameter λ_k. Consider the log-likelihood function for n probes to pair $\langle 2, 3\rangle$:

$$
\begin{aligned}
(4.13) \qquad l(\vec{\lambda}; \mathbf{Y}) &= n\log(\lambda_1) + n\log(\lambda_2) + n\log(\lambda_3) - n\log(\lambda_1 - \lambda_2 - \lambda_3) \\
&- \lambda_2 \sum_{i=1}^{n} y_{i,2} - \lambda_3 \sum_{i=1}^{n} y_{i,3} \\
&- \sum_{i=1}^{n} \log[1 - \exp\{-(\lambda_1 - \lambda_2 - \lambda_3)\min(y_{i,2}, y_{i,3})\}]
\end{aligned}
$$

There is no analytic solution to maximize this equation over $\vec{\lambda}$. An iterative technique like EM or Newton-Raphson is required to maximize the likelihood.

As the tree grows, the problem becomes practically intractable. The number of components in the convolution increases and the complete likelihood becomes unwieldy. The maximum likelihood setting becomes impractical in terms of the effort required from the both the statistician and the computer.

Further problems arise when considering other distributions with more general shapes. If each of the link delays follows a gamma distribution with parameters α_k and β_k, then there is no closed form for the likelihood on even the simplest tree. The only obvious solution at this point would seem to involve a grid search combined with numerical integration to evaluate the likelihood at each point.

4.3 Moment Estimation

Because of the difficulties presented by maximum likelihood, we need to pursue other avenues of estimation. Moment estimation provides an alternative scheme. We can choose parameters in order to match observed and fitted moments as closely as possible. This section builds the framework for this procedure.

4.3.1 Problem Formulation

We begin with a loss function. Although there are many choices, we favor squared-error loss. We choose estimation to minimize the squared distance between the observed and fitted moments. The choice is appealing for a number of reasons. First, for large samples, the moments should be approximately normal, thus squared-error loss is appropriate for approximating a normal likelihood. Further, we can borrow from the well-developed literature on optimization in the squared-error setting.

We consider the loss function for some tree with probing experiment $\mathcal{C}$. Let $\mathcal{M}_i^j(\theta)$ be the functional form of the i-th moment from the j-th probing scheme. This may take any of a number of forms. It might be an end-to-end mean, variance, or other higher order central moment. It might be some order of central cross-moment between some collection of the receiver set for $\mathcal{C}_j$. It might also be something like the probability of seeing zero delay or loss along a path or set of paths. All that is required in the subsequent development is that it be some observed moment from end-to-end measurements that converges in probability to its true value and converges in distribution to a normal variate (after appropriate scaling).

As an example, consider the observed values that could be used to fit the following distribution to each link on two-layer, two-leaf tree:

$$P\{X_k = 0\} = p_k, \tag{4.14}$$

$$X_k | X_k > 0 \sim Gamma(\alpha_k, \beta_k). \tag{4.15}$$

Estimation can be performed based on the following end-to-end statistics:

$$\log[P\{\vec{Y} = (0,0)\}] = \log(p_1) + \log(p_2) + \log(p_3) \tag{4.16}$$

$$\log[P\{Y_2 = 0\}] = \log(p_1) + \log(p_2) \tag{4.17}$$

$$\log[P\{Y_3 = 0\}] = \log(p_1) + \log(p_3) \tag{4.18}$$

$$E(Y_2) = (1 - p_1)\alpha_1\beta_1 + (1 - p_2)\alpha_2\beta_2 \tag{4.19}$$

$$E(Y_3) = (1 - p_1)\alpha_1\beta_1 + (1 - p_3)\alpha_3\beta_3 \tag{4.20}$$

$$Cov(Y_2, Y_3) = p_1(1 - p_1)^2\alpha_1^2\beta_1^2 + (1 - p_1)\alpha_1\beta_1^2 \tag{4.21}$$

$$E(Y_2 - \nu_2)^2(Y_3 - \nu_3) = p_1(1 - p_1)^3\alpha_1^3\beta_1^3 + 2(1 - p_1)\alpha_1\beta_1^3 \tag{4.22}$$

$$Var(Y_2) = \sum_{k=[1,2]} p_k(1 - p_k)^2\alpha_k^2\beta_k^2 + (1 - p_k)\alpha_k\beta_k^2 \tag{4.23}$$

$$Var(Y_3) = \sum_{k=[1,2]} p_k(1 - p_k)^2\alpha_k^2\beta_k^2 + (1 - p_k)\alpha_k\beta_k^2 \tag{4.24}$$

Note that a point mass at infinity can be added by treating all of the above end-to-end statistics as conditional on being finite and adding three more probability-based moments to cover the observations being infinite (or finite as preferred).

The loss function is given by:

$$Q(\theta; \mathbf{M}) = \sum_{j=1}^{|\mathcal{C}_j|} \sum_i \left[M_i^j - \mathcal{M}_i^j(\theta)\right]^2, \tag{4.25}$$

where M_i^j is the observed i-th moment for the j-th scheme.

We note that this setting allows for quite general estimation. Parametric distributions are of course easily fit with this method. Additionally, we can take a semiparametric approach by specifying the functional form for any desired set of moments. Further shape restrictions can be imposed as desired. Finally, as we have seen, this procedure allows us to easily mix a continuous distribution with point masses at zero and infinity. This is appealing for network tomography since it allows to estimate the probability of an empty queue and a full queue along with the queuing distribution for the intermediate case.

4.3.2 Fitting

Because of the similarity to traditional nonlinear least squares, we consider model fitting using the Gauss-Newton procedure (see Bates and Watts (1988) for example). The algorithm is attractive because of simplicity. We develop it here in the context of the moment estimation. Throughout, consider the simple example of the two-layer, two-leaf tree with exponential delay distributions on each link. The model yields the following vectorized end-to-end moments:

$$\mathcal{M}(\theta) = \begin{bmatrix} \theta_1 + \theta_2 \\ \theta_1 + \theta_3 \\ \theta_1^2 \\ \theta_1^2 + \theta_2^2 \\ \theta_1^2 + \theta_3^2 \end{bmatrix}. \tag{4.26}$$

The entries are the means, the covariance, and the variances of the end-to-end values.

Rewrite the loss function as one sum over all the moments and consider its derivatives:

$$Q(\theta; \mathbf{M}) = \sum_i \left[M_i - \mathcal{M}_i(\theta)\right]^2, \tag{4.27}$$

$$\frac{\partial Q(\theta; \mathbf{M})}{\partial \theta_j} = -2 \sum_i \left[M_i - \mathcal{M}_i(\theta)\right] \frac{\partial \mathcal{M}_i(\theta)}{\partial \theta_j}. \tag{4.28}$$

We can consider all of the derivatives in matrix form:

$$\left[\frac{\partial Q(\theta; \mathbf{M})}{\partial \theta}\right] = D'[M - \mathcal{M}(\theta)], \tag{4.29}$$

where

$$D_{i,j} = \frac{\partial \mathcal{M}_i(\theta)}{\partial \theta_j}. \tag{4.30}$$

For the simple example, we have the following matrix of partial derivatives:

$$D = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 2\theta_1 & 0 & 0 \\ 2\theta_1 & 2\theta_2 & 0 \\ 2\theta_1 & 0 & 2\theta_3 \end{bmatrix} \tag{4.31}$$

In order to minimize the least-squares criterion, we need to find θ such that the above equation is equal to a vector of zeros. Since an analytic solution is not available, we turn to the Gauss-Newton search procedure.

We can approximate the moments at the true value using a Taylor expansion around some initial guess $\theta^{(0)}$:

$$\mathcal{M}(\theta_0) = \mathcal{M}(\theta^{(0)}) + D(\theta_0 - \theta^{(0)}), \tag{4.32}$$

Forming the residuals and replacing the true value with the observed moments gives us an updating scheme based on solving a linear system. Thus at some iteration q, we have the following linear system.

$$M - \mathcal{M}(\theta^{(q)}) = D\beta. \tag{4.33}$$

We can solve the system and get the next iteration given by $\theta^{(q+1)} = \theta^{(q)} + \hat{\beta}$.

In general each iteration should be closer to the minimizer of the loss function. However, there may be situations where the step increases the sum of squares. To avoid this, we use the modified Gauss-Newton in which the next iteration is given by $\theta^{(q+1)} = \theta^{(q)} + r\hat{\beta}$ where $0 < r \leq 1$. This fraction can be chosen adaptively at each step. If the full step reduces the sum of squares, then it is taken. Otherwise, set $r = .5$. If the half step fails to reduce the sum of squares, then it is halved again. This guarantees that the loss function is reduced with every step. This guarantees convergence to a stationarity point. Examination of the derivatives will indicate if the point is a minimum.

4.3.3 Large Sample Behavior

The procedure has desirable asymptotic properties that allow for inference in the large sample setting.

Proposition IV.3. *Let $\mathcal{T}$ be a general tree network. Let $\mathcal{C}$ be a collection of k-cast schemes $\mathcal{C}_j,\ j = 1, ..., C$. Let each link delay distribution be given by F_k. Assume that the model specification is correct and that the F_k are identifiable from $\mathcal{C}$. The least-squares moment estimator is consistent and asymptotically normal:*

$$\hat{\theta} \;\rightarrow\; \theta_0 \text{ w.p. } 1 \tag{4.34}$$

$$\sqrt{\mathbf{n}}(\hat{\theta} - \theta_0) \;\Rightarrow\; Z \sim N(\vec{0}, D_\dagger' \Sigma D_\dagger), \tag{4.35}$$

where $D_\dagger = (D'D)^{-1}D'$ and D is the matrix of partial derivatives and Σ is the covariance of the observed moments.

Proof: We begin with consistency. Consider the vector of partial derivatives of the loss function set equal to zero:

$$\left[\frac{\partial Q(\theta; \mathbf{M})}{\partial \theta}\right] = \vec{0}. \tag{4.36}$$

If the model is correct, then one solution of this equation is θ_0 and $\mathcal{M}(\theta_0)$. Further, the Hessian at this value is given by: $H = D'D$ where D is the matrix of partial derivatives. If the model is identifiable, then the columns of D are linearly independent and H is strictly positive-definite. By the implicit function theorem, there is a neighborhood U around $\mathcal{M}(\theta_0)$, a neighborhood V around θ_0, and a unique, continuously differentiable function $\varphi : U \rightarrow V$ such that

$$\left[\frac{\partial Q(\theta; M)}{\partial\theta}\right]_{\theta=\varphi(M)} = \vec{0}, \forall M \in U. \tag{4.37}$$

By the Strong Law of Large Numbers, M will be in U almost surely. Thus the minimizer $\hat{\theta}$ that makes $Q(\theta, M) = 0$ is consistent.

Given consistency, asymptotic normality is relatively straightforward. Consider the Taylor expansion of the observed moments around the true value. Note for large samples $\hat{\theta}$ sets the loss function equal to zero. In the following, h_i is the Hessian for the i-th moment evaluated at a point between $\hat{\theta}$ and θ_0 in order to give equality. Here $\mathbf{n}$ is a diagonal matrix formed from the sample size for each scheme.

$$M = \mathcal{M}(\theta_0) \quad + \quad D(\hat{\theta} - \theta_0) + \frac{1}{2}\begin{pmatrix} (\hat{\theta} - \theta_0)' h_1 (\hat{\theta} - \theta_0) \\ \vdots \\ (\hat{\theta} - \theta_0)' h_m (\hat{\theta} - \theta_0) \end{pmatrix} \tag{4.38}$$

$$[M - \mathcal{M}(\theta_0)]\sqrt{\mathbf{n}} \quad = \quad D(\hat{\theta} - \theta_0)\sqrt{\mathbf{n}} + \frac{1}{2}\begin{pmatrix} (\hat{\theta} - \theta_0)' h_1 (\hat{\theta} - \theta_0)\sqrt{\mathbf{n}} \\ \vdots \\ (\hat{\theta} - \theta_0)' h_m (\hat{\theta} - \theta_0)\sqrt{\mathbf{n}} \end{pmatrix} \tag{4.39}$$

As the sample size goes to infinity, the term on the right is distributed as normal variate with mean $\vec{0}$ and covariance Σ. The Hessians are bounded on the closure of V because they are continuous. As a result, the last term goes to zero as $\hat{\theta}$ goes to

θ_0. Rearranging terms, followed by an application of the delta method gives us

$$(\hat{\theta} - \theta_0)\sqrt{\mathbf{n}} \Rightarrow N(\vec{0}, D_\dagger' \Sigma D_\dagger).\square \tag{4.40}$$

With these facts, we can compute approximate confidence regions and hypothesis tests, key components for monitoring applications.

4.3.4 Semiparametric Estimation

Specifying a complete distribution can be quite restrictive and there may be no obvious choice. As a result, we also consider semiparametric model fitting in which moments are specified as desired. These can be used to characterize the distributions under study or as a guide toward choosing appropriate parametric models.

In this section, we will develop the following model in detail:

$$\mathrm{E}(X_k) = \mu_k, \tag{4.41}$$

$$\mathrm{Var}(X_k) = \phi\mu_k^\gamma. \tag{4.42}$$

This model, partially inspired by generalized linear modeling and quasi-likelihood modeling, has a number of appealing features: it is fairly simple with a form that is similar to many common parametric distributions including exponential, gamma, and log-normal. A similar model was used in Cao et al. (2000) for packet count data in the passive tomography setting where the packet counts were modeled as normally distributed with variance proportional to the mean raised to a power. Accordingly, the model is appropriate for delays since they can act as a proxy for packet counts. This model is easily extended to include point masses at zero (representing an empty queue) and infinity (representing a full queue and a lost packet). Thus, our complete model becomes:

$$P\{X_k < \infty\} = \alpha_k \tag{4.43}$$

$$P\{X_k = 0 | X_k < \infty\} = p_k \tag{4.44}$$

$$E\{X_k | 0 < X_k < \infty\} = \mu_k \tag{4.45}$$

$$Var\{X_k | 0 < X_k < \infty\} = \phi\mu_k^\gamma. \tag{4.46}$$

First, we verify sufficient conditions for the estimability of this model. For the model with the point masses, an earlier condition applies and the model is estimable. In the following proposition, we consider only the purely continuous model.

Proposition IV.4. *Let $\mathcal{T}$ be a general tree network, and suppose its link delay distributions are of the form in Equation 4.41. Let $\mathcal{C}$ be a collection of flexicast schemes. The parameters of the link-level distributions are identifiable if: (a) For every internal node $s \in \mathcal{I}$, there is at least one scheme $\mathcal{C}_j \in \mathcal{C}$, with at least two receivers such that s is a branching node for $\mathcal{C}_j$, (b) every receiver $r \in \mathcal{R}$ is covered by at least one $\mathcal{C}_j \in \mathcal{C}$, (c), the μ_k are distinct, and (d) $\gamma \neq 0$.*

Before beginning the proof, we consider the simplest example: the two-layer, two-leaf tree. If we attempt to estimate the model using the Gauss-Newton search to minimize squared-error, we would have the following matrix of partial derivatives:

$$D = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ \phi\gamma\mu_1^{\gamma-1} & 0 & 0 & \mu_1^\gamma & \phi\log(\mu_1)\mu_1^\gamma \\ \phi\gamma\mu_1^{\gamma-1} & \phi\gamma\mu_2^{\gamma-1} & 0 & \mu_1^\gamma + \mu_2^\gamma & \phi\log(\mu_1)\mu_1^\gamma + \phi\log(\mu_2)\mu_2^\gamma \\ \phi\gamma\mu_1^{\gamma-1} & 0 & \phi\gamma\mu_3^{\gamma-1} & \mu_1^\gamma + \mu_3^\gamma & \phi\log(\mu_1)\mu_1^\gamma + \phi\log(\mu_3)\mu_3^\gamma \end{bmatrix}. \tag{4.47}$$

The model is identifiable when the columns of the partial derivative matrix are linearly independent. For small examples, it is easy to see that this condition is satisfied whenever the μ_k are distinct. Examining this matrix, we can see that when $\mu_k = \mu$, there is no way to estimate the parameters using this technique.

Remark 1: As we shall see, the restriction on the μ_k arises for technical reasons, but the case $\mu_k = \mu$ is exceptional for other reasons. If all of the links have the same mean the estimation procedure is trivial: probe the nearest receiver to the source and determine the link means based on the path mean and the number of links. As we shall see, if the entire tree is probed, the data can be used to diagnose identical means in which case we can use an alternative estimator.

Proof: First note that it is easy to obtain each link variance from the end-to-end covariance matrices. Since there is some scheme with receivers k and l that split at node 1, the covariance of the data received at nodes k and l for this scheme gives us the variance of link 1. All of the rest of the variances can be obtained by peeling.

Let the variance of link k be denoted v_k and let m be the link with the smallest variance. Let us define the following variance ratios:

$$\rho_k = \frac{v_k}{v_m}. \tag{4.48}$$

These quantities are all greater than or equal to one. Note also that

$$\rho_k^{\frac{1}{\gamma}} = \frac{\mu_k}{\mu_m}. \tag{4.49}$$

Denote the end-to-end mean at receiver k as ν_k. The ratio of two end-to-end means can be written:

$$\frac{\nu_k}{\nu_l} = \frac{\sum_{i \in \mathcal{P}_{\prime,\|}} \rho_i^{\frac{1}{\gamma}}}{\sum_{i \in \mathcal{P}_{\prime,\|}} \rho_i^{\frac{1}{\gamma}}}. \tag{4.50}$$

Moving the left term to the right gives us a function of γ. If the ratio of end-to-end variances is one, then the function is identically zero or has an horizontal asymptote at zero and no solution can be found. Since the the μ_k are distinct by assumption, there is at least one ratio of end-to-end means that is not identically zero. This restriction can be relaxed: an appropriate end-to-end mean ratio can be found if the receiver links are all distinct. Otherwise, the function is one-to-one with a zero at the true value of γ so this parameter is identifiable.

Given γ, we can easily solve for μ_m and thus all of $\vec{\mu}$ based on the end-to-end means. Given γ and $\vec{\mu}$, we can solve for ϕ, from the variances and the parameters are identified. □.

To clarify the proof, consider the simple example of the two-layer, two-leaf tree. Suppose that $\vec{\mu} = [3, 2, 4]$, $\phi = 1.5$, and $\gamma = 2$. We get the following end-to-end statistics: $\vec{\nu} = [5, 7]$, $Var(Y_2) = 19.5$, $Var(Y_3) = 27.5$, and $Cov(Y_2, Y_3) = 13.5$. First, from the end-to-end variance information, we know that $\vec{v} = [13.5, 6, 24]$ and thus $\vec{\rho} = [2.25, 1, 4]$. We can now form the appropriate function of γ:

$$f(\gamma) = \frac{2.25^{1/\gamma} + 1^{1/\gamma}}{2.25^{1/\gamma} + 4^{1/\gamma}} - \frac{5}{7}. \tag{4.51}$$

Consider the plot of this function in Figure 4.1. There is clearly a single zero at the true value. The plot demonstrates the typical behavior of these functions. With γ correctly estimated, the other parameters are easily obtained.

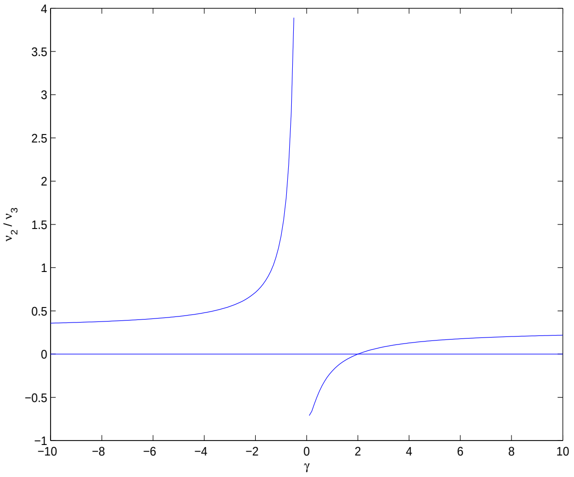

Figure 4.1: The function of gamma based on end-to-end statistics with the x-axis. There is one zero at the correct value.

To see how the potential nonidentifiable scenarios arise, consider two examples. First, consider the two-layer, two-leaf tree with the semiparametric model with all means the same. It is easy to see that $\theta_0 = [\mu_0, \phi_0, \gamma_0]$ gives the same end-to-end statistics as $\theta_1 = [\mu_0, \mu_0^{\gamma_0 - 1}, 1]$. Further, if $\mu = 1$, then neither ϕ nor γ can be estimated.

Second consider the same tree with $\vec{\mu} = [3, 2, 2]$, $\phi = 1.5$, and $\gamma = 2$. The end-to-end means are $\vec{\nu} = [5, 5]$ and the link variances are $\vec{v} = [9, 4, 4]$. Using these statistics and assuming $\gamma = 1$, we can arrange the following equations:

$$\phi\mu_1 = 9 \tag{4.52}$$

$$\phi(5 - \mu_1) = 4. \tag{4.53}$$

Solving gives $\mu_1 = 45/13$ and $\phi = 13/5$.

In both cases, the relationship among the μ_k effectively limits the number of distinct estimating equations preventing distinct estimation.

Because of the potential pitfall, we need a method for diagnosing the problem beforehand so that other techniques can be used if necessary. Additionally, we will need to modify the Gauss-Newton search to avoid trouble areas of the loss function.

Diagnosing Nonidentifiability

If the end-to-end means are all identical, then the model as stated cannot be estimated. Troublesome links can be determined by considering the link variances which can be found by manipulating the end-to-end covariance matrix.

In practice, the model can often be fit even when the underlying model is troublesome. The error observed in the moments is usually sufficient to offset the estimates enough for the model to be fit. The difficulty only seems to arise for extremely accurate observed moments when the underlying model is bad.

Finally, when the end-to-end means do indicate a problem, the model can still be estimated, but the appropriate restrictions must be imposed. Equal means must be represented as a single parameter rather than individually. Further, it may be necessary to fix either ϕ or γ. For example, if γ is fixed at some nominal or assumed value, than the rest of the parameters are easily estimable.

Block-Relaxed Gauss-Newton

The nonidentifiability can cause the algorithm to fail even when the true model is identifiable. If the search procedure attempts to evaluate a troublesome point, the matrix $D'D$ becomes singular and the algorithm breaks down. In order to alleviate this, we propose a modification to the search procedure in which $\vec{\mu}$ and ϕ are updated separately from γ.

Partition the partial derivative matrix into two components: $D = [D_{\vec{\mu},\phi}, D_\gamma]$. Algorithmically, we are simply doing two updates: one using $D_{\vec{\mu},\phi}$ and one using D_γ. Since any singularity in the $D'D$ matrix arises because of the γ column, this partitioning avoids any difficulty. We know that $\vec{\mu}$ and ϕ can be estimated when γ is known so the columns in the first partition will be independent for any value of the γ so $D'_{\vec{\mu},\phi}D_{\vec{\mu},\phi}$ will always be invertible. Each iteration of the algorithm now has two steps: (1) compute the update $\hat{\beta}_{\vec{\mu},\phi}$ and adjust both the appropriate parameters and the fitted values and (2) compute the update $\hat{\beta}_\gamma$ and adjust γ and the fitted values again. If necessary, further blocking can be used to avoid trouble.

Convergence results under this partitioned algorithm are much the same as in the general case. Modifying the step size to reduce the sum of squares gives us convergence to some critical value.

4.3.5 Weighting

In the traditional nonlinear least squares setting, observations with different variances can be weighted to assist estimation. The same procedure can be used here in the moment matching setting, particularly as the variance for moments is often well-known. Two procedures can be used depending on the setting. If the entire distribution is specified and the variance of each moment can be calculated in terms of the parameters, then the weights can be updated with each new iteration of the parameters. Alternatively, the variance of each moment can be computed based on the observations and held constant throughout the estimating procedure. The weighting generally helps to speed convergence and does not affect the other properties of the algorithm.

4.4 Independence Violations

The focus of this thesis is on the spatio-temporal independence model. Unfortunately, we do not live in this world and it is worth investigating the performance of the estimation when the scenario violates the assumption. In Chapter III, the discrete model was successfully applied to ns-2 simulated data suggesting that it is fairly robust to these violations. The moment techniques are inherently more vulnerable since a great deal of data aggregation takes place before estimation. This section examines the effect of these violations and offers some discussion of solutions.

4.4.1 Spatial Dependence

If two links share a significant amount of the same traffic, they will exhibit a positive correlation. Further, if two links share a common parent, then processes on the parent, such as TCP traffic regulation, can impart a spatial correlation structure to all three links. Spatial dependence can have a particularly adverse effect on the moment estimation techniques. If the delays along the branches of a probing tree are positively correlated, the covariance of the end-to-end results is inflated beyond just the variance of the shared path. The other cross-moments will be similarly effected. Since the relationship among higher order moments is used to assign the end-to-end mass, the inflated cross moment will tend to shift most of the delay to the shared links.

Consider the simple two-layer, two-leaf tree with a simple spatial structure in which the delay on a link is correlated with the delays on its children. For example, we could have the following exponential model:

$$\epsilon_k + 1 \sim Exp(1), \tag{4.54}$$

$$Z_1 = \epsilon_1, \tag{4.55}$$

$$Z_2 = (1-\rho)\epsilon_2 + \rho\epsilon_1, \tag{4.56}$$

$$Z_3 = (1-\rho)\epsilon_3 + \rho\epsilon_1, \tag{4.57}$$

$$X_k = \theta_k + \theta_k Z_k. \tag{4.58}$$

The admittedly strange structure is require to keep all of the delays positive while still imposing some spatial dependency. When $\rho = 0$, this reduces to the independent exponential model.

Figure 4.2 shows the bias for θ_1 when trying to estimate the above model using just an independent exponential framework. The results are based on simulations of the above model with $\theta_k = 1$ and ρ ranging from 0 to .9. As expected, for large spatial dependency, the link 1 parameter is overestimated by almost 100%. The linear trend implies that even smaller dependencies have an immediate effect on estimation.

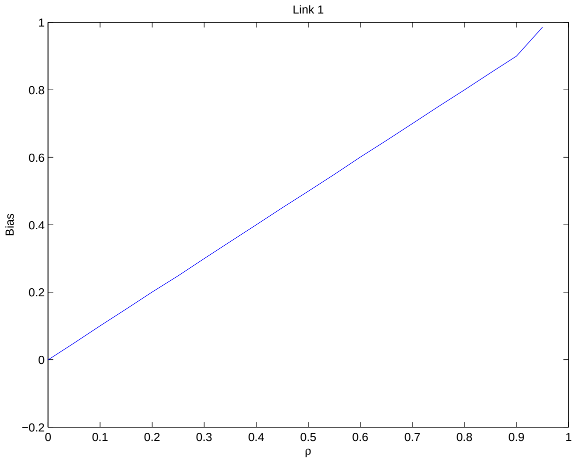

Figure 4.2: Bias for link 1 for increasing spatial dependencies.

A Priori Correction

If the dependence structure and parameters are known, the problem can usually be easily solved. For the above scenario, consider the end-to-end moments:

$$E(Y_2) = \theta_1 + \theta_2, \tag{4.59}$$

$$E(Y_3) = \theta_1 + \theta_3, \tag{4.60}$$

$$Var(Y_2) = \theta_1^2 + 2\rho\theta_1\theta_2 + (1 - 2\rho + 2\rho^2)\theta_2^2, \tag{4.61}$$

$$Var(Y_3) = \theta_1^2 + 2\rho\theta_1\theta_3 + (1 - 2\rho + 2\rho^2)\theta_3^2, \tag{4.62}$$

$$Cov(Y_2, Y_3) = \theta_1^2 + \rho\theta_1\theta_2 + \rho\theta_1\theta_3 + \rho^2\theta_2\theta_3. \tag{4.63}$$

If ρ is known, then estimating this model is no more difficult than estimating the independence model. Further, the model can be extended easily to the case in which X_2 and X_3 have differing levels of correlation with their parent or even the case in which $\vec{X}$ has an unrestricted correlation structure. Other stochastic models may be used as well although the appropriate moments will have to be considered.

Full Estimation

If the dependence structure is unknown, the situation is somewhat more difficult. Even if the simple case above, we are now interested in more parameters and identifiability issues come into play. Further, determining an appropriate spatial correlation structure from end-to-end data can be very difficult and fitting an unrestricted spatial model is not always feasible.

By example, consider the above model. A model of this form seems to be identifiable. If we repeat the simulation used to obtain the bias results but instead fit the moment model based on the described data, we can fit the model correctly. Figure 4.3 shows the bias for link 1 under this model. Clearly the bias is almost zero for all by the largest value of ρ and even here it is relatively small. Indeed, the model is still asymptotically consistent so for any given ρ, the bias will go to zero as the sample size increases.

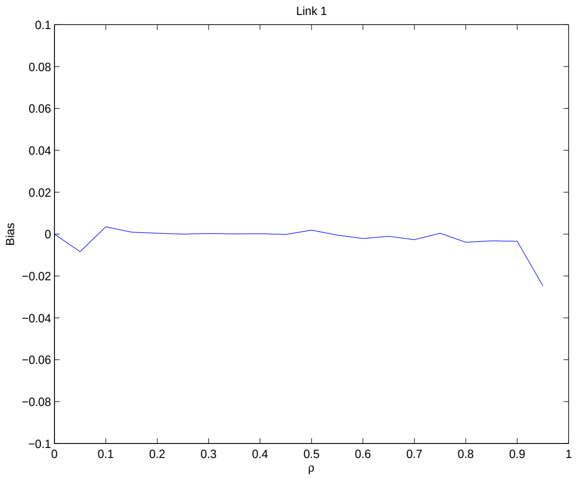

Figure 4.3: Bias for link 1 under the spatial model for increasing spatial dependencies.

In general, the estimation framework that we describe can be extended as long as identifiable models are produced. As is generally the case, additional parameters may require the consideration or specification of additional moments. Further work regarding appropriate models and their identifiability is currently ongoing.

4.4.2 Temporal Dependence

Temporal dependence on a link arises because consecutive probes see more similar network conditions than do disparate probes. Depending on the network, this effect can be quite strong. As in the spatial setting, this correlation can cause some moments to appear inflated in relation to others, thus skewing our estimation technique. To examine the effects of this, consider a time-series model similar to the spatial model given above:

$$\epsilon_k(i) \sim Exp(1), \tag{4.64}$$

$$Z_k(i) = \left(\sum_{j=0}^{L} \rho^j \epsilon_k(i-j)\right) / \sum_{j=0}^{L} \rho^j, \tag{4.65}$$

$$X_k(i) = \theta_k + \theta_k Z_k(i), \tag{4.66}$$

where L is the maximum lag. Again, for $\rho = 0$, the model reduces to the independent exponential model.

Figures 4.4 and 4.5 show the bias on link 1 when estimating the above model with $\theta_k = 1$ using an independent exponential framework. For a given lag, the mean of link 1 is underestimated as ρ increases. For short lags, the underestimate seems to level off at some value while for larger lags, the underestimate appears to reach a maximum and then reduces to some fixed point. For fixed value of ρ, the underestimate of link 1 increases with lag to a fixed point causes by the exponentially decaying weights in our model.

We note that temporal dependence can be controlled to some degree based on the probing interval. Widely spaced probes will experience smaller autocorrelation than back-to-back probes.

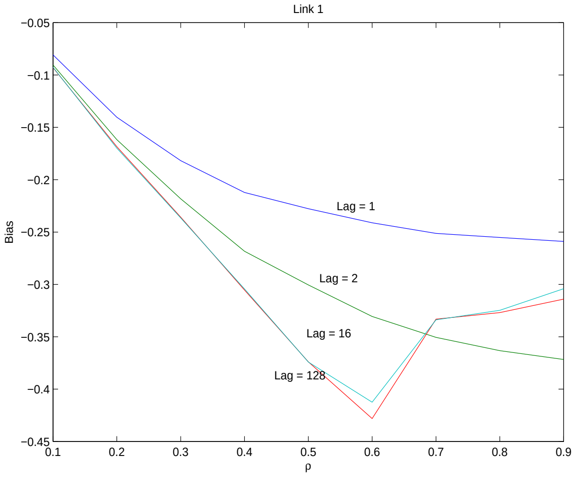

Figure 4.4: Bias for link 1 for increasing values of ρ at varying lags.

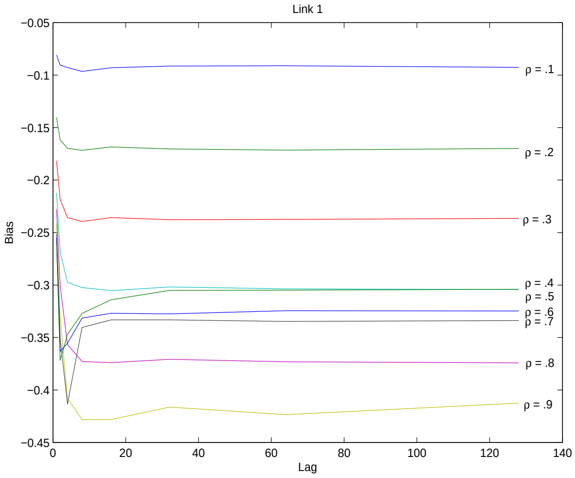

Figure 4.5: Bias for link 1 for increasing lags at varying ρ.

A Priori Knowledge

As in the spatial case, if the structure and parameters of the temporal dependence are known, then it often simple to incorporate the knowledge into the estimation framework. Once again, the least-squares moment matching in this case will usually have the same requirements and conditions as the independent case, but the moments will now have more terms or terms with different coefficients.

In this case, some estimate of the structure and parameters may be gained from the end-to-end data. Autocorrelation in the end-to-end data can be used to obtain an estimate of the autocorrelation for the link data. If the correlation structure can be estimated this way, it can be treated as fixed for the estimation of the rest of the parameters using least-squares.

Full Estimation

Once again, a simple model like that described can be estimated using least-squares without prior knowledge of the parameters. For any model, identifiability must be checked and the appropriate moment computed.

4.5 Numerical Investigation

In this section, we investigate the numerical performance and properties of the techniques that we have described.

4.5.1 Statistical Efficiency: Moment Methods Versus MLE

Here we consider the efficiency of the moment procedures as compared with maximum likelihood. As discussed, it is difficult in general to compute the maximum likelihood estimate so we investigate relatively simple scenarios.

On a two-layer, two-leaf tree, we consider poisson and exponentially distributed link delays. For each scenario, estimation is performed on 1000 data sets of 1000 observations each. For both distributions, the mean of links 1, 2, and 3 were set at 1/3, 1/2, and 1 respectively. For each data set, we consider the maximum likelihood estimator, the parametric moment estimator, the semiparametric estimator with known ϕ and γ, and the semiparametric estimator with all parameters unknown. Figures 4.6, 4.7, and 4.8 show the results for the poisson data and Figures 4.9, 4.10, and 4.11 show the results for the exponential data. Tables 4.1 and 4.2 show the relative efficiency for the moment estimators as compared with the maximum likelihood estimators. For the poisson data, the moment estimators are about 50%

	Link 1	Link 2	Link 3
Parametric Moment	1.56	1.24	1.33
Semiparametric (Means Only)	1.56	1.28	1.23
Semiparametric	1.47	1.40	1.28

Table 4.1: Relative efficiency as compared with MLE for the poisson data.

	Link 1	Link 2	Link 3
Parametric Moment	1.82	1.57	1.82
Semiparametric (Means Only)	1.77	1.53	1.20
Semiparametric	2.52	2.23	1.46

Table 4.2: Relative efficiency as compared with MLE for the exponential data.

less efficient than the MLE and for the exponential data, the moment estimators are between 60% and 130% less efficient. This suggests that as the variance depends on larger powers of the mean, the accuracy of the moment estimators begins to suffer. It is also interesting to note that the semiparametric and parametric models perform about the same. In both of these cases, the semiparametric moments fit the true moments perfectly in the sense that ϕ and γ are indeed constant over different distributions. Further, for the parametric moment modeling, the algorithms used only the end-to-end means and covariances, the same data that the semiparametric model uses. The parametric assumption would allow us to specify more moments if desired and this may have a constraining effect on the estimates leading to more accurate results.

4.5.2 Statistical Efficiency: Parametric Versus Semiparametric

In this subsection, we consider the efficiency of the semiparametric estimator versus the efficiency of the appropriate parametric estimator for moment matching. We conduct a limited simulation study to assess the performance.

Data were generated three-, four-, and five-layer binary, symmetric trees with exponential delay distributions on each link. The mean of the root link is always three, the means of the internal links are all two, and the means of the receivers links

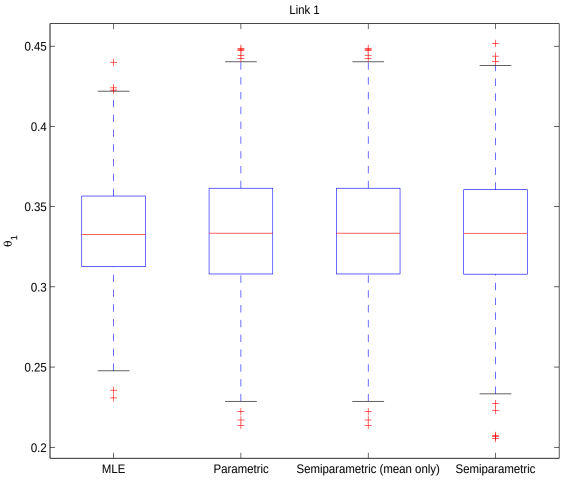

Figure 4.6: Boxplot of estimated means on link 1 for the MLE, parametric moment estimator, and the semiparametric moment estimator with and without known ϕ and γ for poisson data.

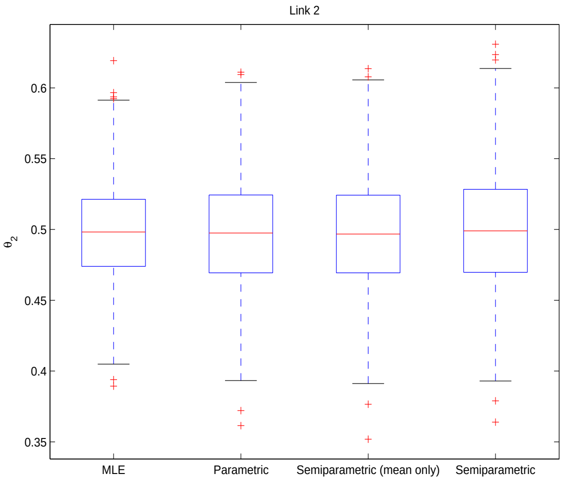

Figure 4.7: Boxplot of estimated means on link 2 for the MLE, parametric moment estimator, and the semiparametric moment estimator with and without known ϕ and γ for poisson data.

are always four. Each data set consists of 10,000 probes sent to each scheme in a set of minimum bicast pairs. For each tree, 1000 data sets were produced. For each data set, we applied both the semiparametric estimator and the exponential moment estimator. Table 4.3 shows the average relative efficiency of the semiparametric estimator to the parametric estimator for mean estimation on each layer of the tree. Overall, the parametric estimator does much better, particularly on the receivers

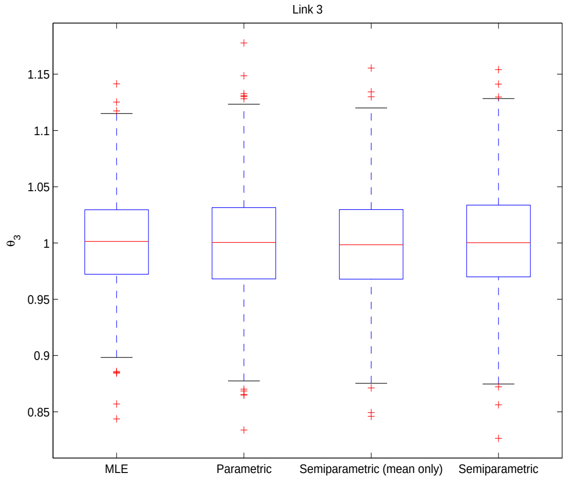

Figure 4.8: Boxplot of estimated means on link 3 for the MLE, parametric moment estimator, and the semiparametric moment estimator with and without known ϕ and γ for poisson data.

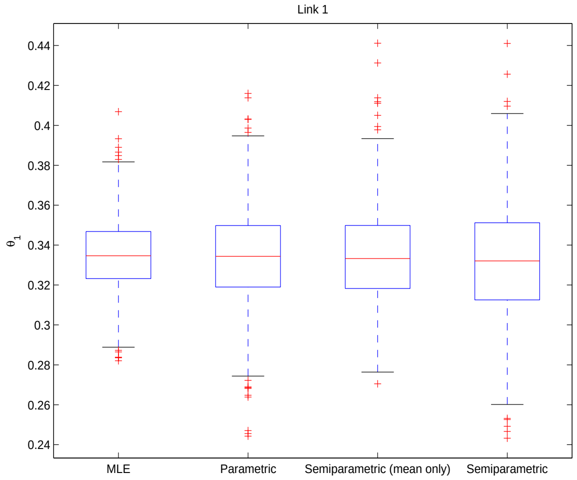

Figure 4.9: Boxplot of estimated means on link 1 for the MLE, parametric moment estimator, and the semiparametric moment estimator with and without known ϕ and γ for exponential data.

where it is more than twice as efficient. Interestingly, the semiparametric estimator outperforms the parametric estimator for the layer immediately above the receiver set. The reason for this likely stems from the fact that the semiparametric estimator is somewhat more flexible than the exponential parametric model. By adjusting the ϕ and γ, it is gaining some improvement in the means at this layer.

A more general investigation should consider a more flexible model than the ex-

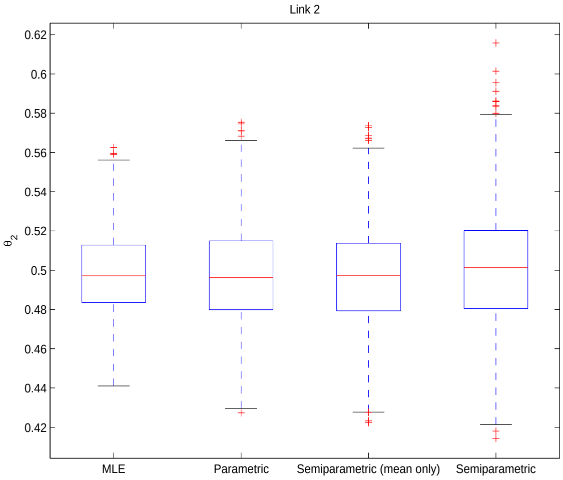

Figure 4.10: Boxplot of estimated means on link 2 for the MLE, parametric moment estimator, and the semiparametric moment estimator with and without known ϕ and γ for exponential data.

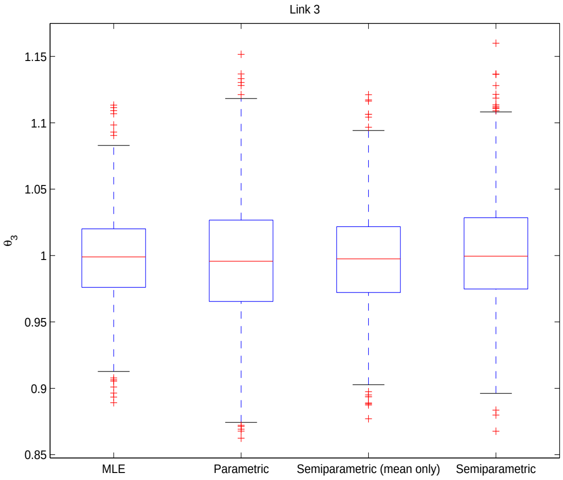

Figure 4.11: Boxplot of estimated means on link 3 for the MLE, parametric moment estimator, and the semiparametric moment estimator with and without known ϕ and γ for exponential data.

ponential. It seems unlikely that the semiparametric model would still outperform a two or more parameter model at any level in the tree. Further, increasing the sample size for larger tree sizes might show that the parametric model outperforms the semiparametric model through faster convergence to a tighter limiting distribution. Finally, although not rigorously studied here, the semiparametric estimation often took considerably longer than the parametric. This is likely a result of the

Number of Layers:	3	4	5
Layer 1 Ratio	1.63	1.72	2.11
Layer 2 Ratio	1.13	1.27	1.45
Layer 3 Ratio	2.38	.86	1.23
Layer 4 Ratio		2.00	.88
Layer 5 Ratio			2.68

Table 4.3: Average relative efficiency of the semiparametric estimator to the parametric estimator for mean estimation on each layer in the tree.

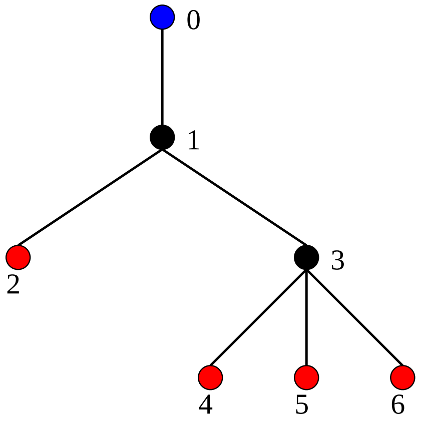

Figure 4.12: Portion of the UNC network.

partitioned updating algorithm, but should be studied further as well.

4.6 Network Simulation

In order to assess the performance under realistic conditions, we apply them to simulated data produced using ns-2. We focus on simulating a portion of the UNC topology seen in Figure 4.12. Because of the sensitivity of the moment methods to violations of the assumptions, we simplify the simulation as compared with that done in the previous chapter. Each link has a bandwidth of 100 Mb and the background traffic, a combination of TCP and UDP similar to the edge links in the discrete example, are the same on all links.

Because we have no previous assumptions about the shape of the distribution, we choose to fit the semiparametric model with point masses at zero. Figure 4.4 shows the observed and fitted values for each of the parameters. The most obvious result is

Parameter	Observed	Estimated
p_1	.17	.16
p_2	.17	.17
p_3	.18	.14
p_4	.17	.27
p_5	.17	.24
p_6	.17	.22
μ_1	$41.79\mu s$	$42.65\mu s$
μ_2	$42.21\mu s$	$40.81\mu s$
μ_3	$41.27\mu s$	$41.65\mu s$
μ_4	$41.92\mu s$	$43.38\mu s$
μ_5	$42.48\mu s$	$41.87\mu s$
μ_6	$41.67\mu s$	$40.15\mu s$
ϕ	.32	.35
γ	2.00	1.99

Table 4.4: Observed and fitted values of the semiparametric parameters for the ns-2 simulation.

the discrepancy with the point masses for links 3, 4, 5, and 6. The probability on the internal link is too low while that of the receiver links is too high. This shifting of mass down the tree is consistent with some sort spatial correlation and an appropriate model would likely correct the problem. The difficulty, as discussed previously, lies in choosing such a model and diagnosing its necessity from the end-to-end results. Surprisingly, this has little effect on the rest of the parameters, especially the means, which are all quite close to their true values. Indeed overall, the model fits quite well despite the independence violations.

CHAPTER V

Conclusion

5.1 Conclusion

We have considered the estimation of link delay distributions on tree-shaped networks under the assumption of temporal and spatial independence. Under this general framework, we have considered identifiability and estimation for both discrete and continuous modeling.

First, we examined a discretized delay model in which the continuous link delay is binned in units of fixed size. We proposed two estimation schemes: maximum likelihood and a local maximum likelihood scheme called grafting. For the model, we established a necessary and sufficient probing condition to ensure that the model can be estimated. We also proved asymptotic inference results for the estimators. Several numerical aspects of the estimation schemes were considered including parallelization and performance under varying conditions. The methods were applied to Voice-Over-IP data.

Secondly, we investigated continuous delay modeling based on moment estimation. Parameters are estimated in order to minimize squared error between observed and fitted moments. Within this framework, we consider several models. Several sufficient conditions are proved that guarantee identifiability. Asymptotic results are proved that allow for approximate inference tests. Again, we investigated some numerical performance in realistic settings.

A main goal of this research is to provide practical tools for the network researcher. Our focus on both fast and efficient modeling provides tools that can be used in a number of ways. Data sets collected at short repeated intervals can be used for monitoring with the faster algorithms. The efficiency of our models allows for good monitoring even when the data sets are small. Both the discrete model with large bins and the continuous framework can be used in this situation. This type of modeling is useful for anomaly detection and routing. Larger, more extensive data sets can be used to assess detailed performance and do capacity planning. The discrete model can be used to obtain detailed nonparametric estimates of the distributions in this case. Depending on the strength of the assumptions, the continuous modeling can also be used. The methods that we have developed would benefit greatly from further refinement, but can already be useful in many situations.

5.2 Future Work

5.2.1 Mixture Modeling

One future research direction is delay modeling using mixture distributions. This flexible class of models can be used to model distributions of many shapes. Mixtures of exponentials can be used to model tail behavior to any degree of accuracy. This framework would allow extremely detailed modeling of upper quantiles, an important quantity for many network applications. Further, mixtures of gamma distributions should be able to approximate any positive-valued distribution to any desired degree of accuracy.

Within this framework, model fitting is quite formidable. In the case of exponential mixtures, it is possible to derive an EM algorithm to fit the maximum likelihood model. As in the discussion of maximum likelihood in Chapter IV, the estimation becomes practically intractable for large trees. Simple estimation procedures for these types of models would be extremely useful.

5.2.2 Spatio-Temporal Models

The current state of the network tomography literature relies on the assumption of spatial and temporal independence. These assumptions are always violated to some extent. The discrete model seems to be fairly robust to violations of these assumptions based on the ns-2 simulations. The moment matching techniques for the continuous delay modeling are inherently vulnerable to violations: spatial and temporal correlations affect the estimation of the moments and must be taken into account.

For loss modeling, latent variable probit models seem to have merit for extending to the dependent scenario. In this framework, the value of the binary variable X is determined by the value of an underlying latent normal variable Z with unknown mean and variance one. When $Z > 0$, $X = 1$. The mean is estimated to match observed probabilities. Typically, the model is used to connect the probability of success/failure to a set of covariates. Building a time series out of the latent variable induces a time series on the binary variables. Using a multivariate probit model with a latent variable for each link allows the estimation of spatial correlations. A multivariate time series would allow the estimation of both types of correlation. In other work, the author has studied multivariate probit models and the techniques could be extended to account for the restrictions that the network's graph structure imposes. The extension to a time series model is still unstudied.

Similar techniques may used to extend the discrete delay model to include spatio-temporal dependence. If each discrete variable is related to a latent continuous variable, then we can follow a similar path to the one described in the loss scenario. The more obvious extension is to build a time series directly on the continuous delays. This would not allow us to extend the well-known models for normal time series, but it is likely a fruitful direction.

5.2.3 Testing: Goodness-of-Fit and Nested Models

This thesis focuses almost entirely on model estimation. Goodness of fit is also important. There are several questions here. First, if a parametric model is chosen, how appropriate is the model? The tree model would seem to prevent the use of a probability plot, but is there some equivalent? Second, a test to compare "nested" models would be useful. For example, if a gamma or lognormal model is fit, can this be reduced to the semiparametric model?

5.2.4 Regression Models

An area of current study is relating link statistics link like mean and variance to various covariates such as bandwidth and probe packet properties like size and inter-gap interval. The moment estimation procedure would seem to easily allow such an extension.

5.2.5 Other Graphs and Graph-Based Applications

Finally, we would like to see the work ported to other graph-based applications. Many problems can be described with these structures and the methods developed in this paper should be useful. Manufacturing, distribution, and certain biological systems are common areas where this structure exists. An important first step is to generalize the work to more general graphs including multisource topologies and graphs with multiple paths from point to point.

Bibliography

Barakat, C., P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski (2003, August). Modeling internet backbone traffic at the flow level. *IEEE Transactions on Signal Processing 51*(8), 2111–2124.

Bates, D. and D. Watts (1988). *Nonlinear Regression Analysis and Its Applications.* John Wiley & Sons.

Cáceres, R., N. G. Duffield, J. Horowitz, and D. F. Towsley (1999, November). Multicast-based inference of network-internal loss characteristics. *IEEE Transactions on Information Theory 45*(7), 2462–2480.

Cao, J., D. Davis, S. Vander Wiel, and B. Yu (2000). Time-varying network tomography: Router link data. *Journal of the American Statistical Association 95*(452), 1063–1075.

Castro, R., M. Coates, G. Liang, R. Nowak, and B. Yu (2004). Network tomography: Recent developments. *Statistical Science 19*(2), 499–517.

Coates, M. J., R. Castro, M. Gadhiok, R. King, Y. Tsang, and R. Nowak (2002). Maximum likelihood network topology identification from edge-based unicast measurements. In *Proceedings of ACM Sigmetrics.*

Duffield, N. G., J. Horowitz, F. Lo Presti, and D. Towsley (2002). Multicast topology inference from measured end-to-end loss. *IEEE Transactions in Information Theory 48*, 26–45.

Duffield, N. G., F. Lo Presti, V. Paxson, and D. Towsley (2001). Inferring link loss using striped unicast probes. In *IEEE Infocom 2001.*

Fletcher, R. (1987). *Practical Methods of Optimization.* John Wiley & Sons.

Gallant, A. R. (1987). *Nonlinear Statistical Models.* John Wiley & Sons.

Guo, D. and X. Wang (2003, August). Bayesian inference of network loss and delay characteristics with applications to tcp performance prediction. *IEEE Transactions on Signal Processing 51*(8), 2205–2218.

Hartley, H. O. and A. Booker (1965). Nonlinear least squares estimation. *The Annals of Mathematical Statistics 36*, 638–650.

Hohn, N., D. Veitch, and P. Abry (2003, August). Cluster processes: A natural language for network traffic. *IEEE Transactions on Signal Processing 51*(8), 2229–2244.

Information Sciences Institute (2004). *The Network Simulator.* Information Sciences Institute. http://www.isi.edu/nsnam/ns/.

Istratescu, V. (1981). *Fixed Point Theory.* D. Reidel Publishing Company.

Jennrich, R. (1969). Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics 40*, 633–643.

Johnson, N. and S. Kotz (1970). *Continuous Univariate Distributions - 1.* John Wiley & Sons.

Liang, G. and B. Yu (2003, August). Maximum pseudo likelihood estimation in network tomography. *IEEE Transactions on Signal Processing 51*(8), 2043–2053.

Lo Presti, F., N. G. Duffield, J. Horowitz, and D. Towsley (2002, December). Multicast-based inference of network-internal delay distributions. *IEEE Transactions on Networking 10*(6), 761–775.

Marchette, David, J. (2001). *Computer Intrustion Detection and Network Monitoring: A Statistical Viewpoint.* Springer-Verlag.

Medina, A., N. Taft, K. Salamatian, S. Bhattacharayya, and C. Diot (2002). Traffic matrix estimation: Existing techniques and new directions. In *ACM SIGCOMM 2002.*

Nowak, R. D. and M. J. Coates (2001). Unicast network tomography. *IEEE Transactions on Information Theory*. Submitted.

Padmananhan, V. N., L. Qiu, and J. Wang (2003). Server-based inference of internet link lossiness. In *IEEE Infocom 2003.*

Rabbat, M., R. Nowak, and M. Coates (2004). Multiple source, multiple destination network tomography. In *IEEE Infocom 2004.*

Shih, M.-F. and A. O. Hero (2003, August). Unicast-based inference of network link delay distributions with finite mixture models. *IEEE Transactions on Signal Processing 51*(8), 2219–2228.

Shih, M. F. and A. O. Hero (2004). Network topology discovery using finite mixture models. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing.*

Tebaldi, C. and M. West (1998, June). Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association 93*(442), 557–573.

Tsang, Y., M. Coates, and R. D. Nowak (2003, August). Network delay tomography. *IEEE Transactions on Signal Processing 51*(8), 2125–2135.

Vardi, Y. (1996, March). Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association 91*(433), 365–377.

Wu, C.-F. (1981). Asymptotic theory of nonlinear least squares estimation. *The Annals of Statistics 9*, 501–513.

Zhang, Y., M. Roughan, C. Lund, and D. Donoho (2003). An information-theoretic approach to traffic matrix estimation. In *ACM SIGCOMM 2003*.

ABSTRACT

Flexicast Network Delay Tomography

by

Earl Lawrence

Chairs: George Michailidis and Vijayan N. Nair

This thesis considers the estimation of link delay distributions based on path-level measurements in communications networks. The data are collected by actively probing the network: traffic is injected on the network and travels from one accessible point (source) to a group of accessible points (receivers). We observe the end-to-end path-level delays of the injected traffic. The statistical challenge is to deconvolve the observed end-to-end distributions to estimate the link-level distributions. We consider networks represented by tree-shaped graphs: directed graphs in which each node has a single parent except for the source or root node. The probing is based on the multicast protocol in which observed end-to-end delays at the receivers have common components arising from the delay along the shared path to the receivers. We consider two situations. The first is based on discretized link delay in which the delay is assumed to occur in units of fixed size. For this scenario, we prove a necessary and sufficient condition for identifiability and develop two estimators: the maximum likelihood estimate based on the EM algorithm and a faster algorithm based on solving for local MLEs. The second scenario is based on assuming continuous delay, possible with point masses at zero and infinity. Within this framework, we consider several models both parametric and semiparametric. We develop estimation based on moment matching under squared-error loss. Identifiability conditions and inference properties are discussed. The techniques are applied to simulated data and real data collected as part of a Voice-Over-IP testbed.